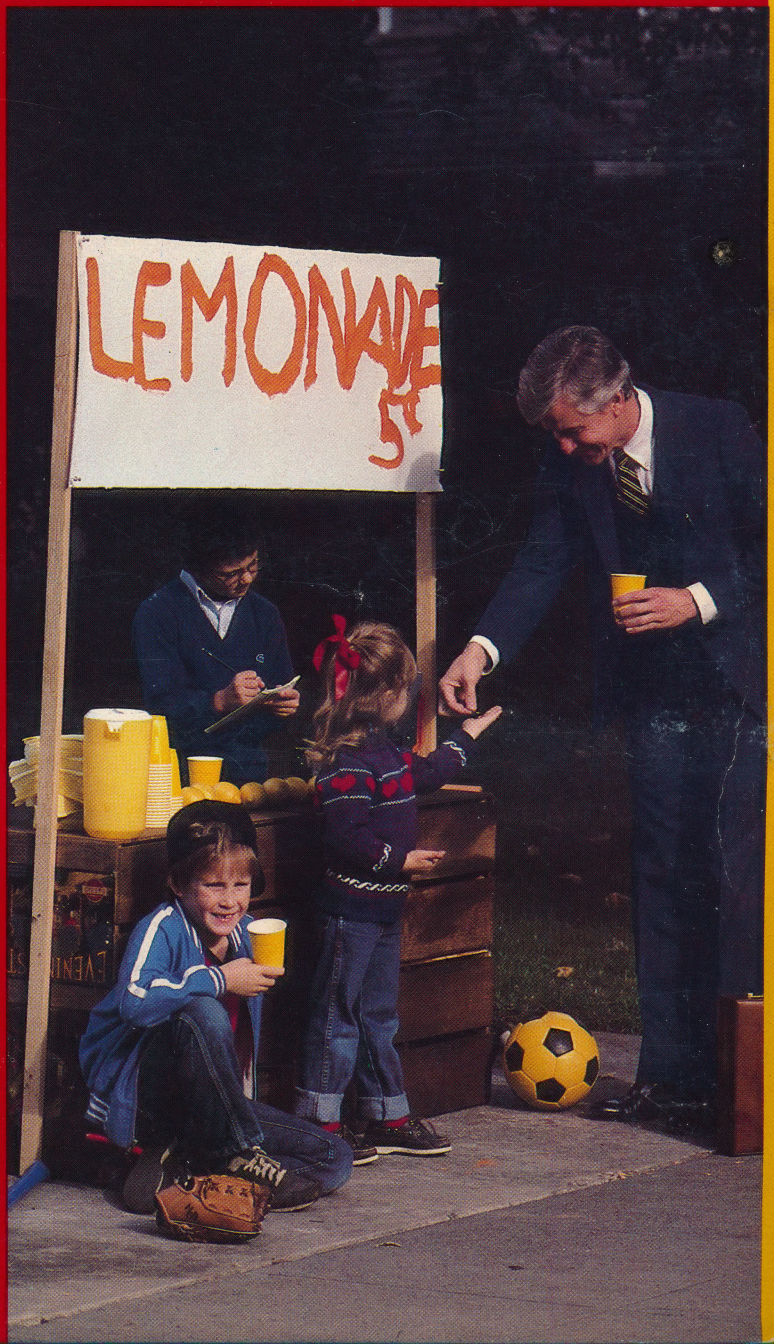
The Apple Education Classics

The Shell Games
Elementary, My Dear Apple

The Apple II



Compatible With
II, II+, IIe, and IIc

Customer Satisfaction

If you discover physical defects in the manuals distributed with an Apple product or in the media on which a software product is distributed, Apple will replace the documentation or media at no charge to you during the 90-day period after you purchased the product.

In addition, if Apple releases a corrective update to a software product during the 90-day period after you purchased the software, Apple will replace the applicable disks and documentation with the revised version at no charge to you during the six months after the date of purchase.

In some countries the replacement period may be different; check with your authorized Apple dealer. Return any item to be replaced with proof of purchase to Apple or an authorized Apple dealer.

Limitation on Warranties and Liability

Even though Apple has tested the software described in the manual and reviewed its contents, neither Apple nor its software suppliers make any warranty or representation, either express or implied, with respect to this manual or to the software described in this manual, their quality, performance, merchantability, or fitness for any particular purpose. As a result, this software and manual are sold "as is," and you the purchaser are assuming the entire risk as to their quality and performance. In no event will Apple or its software suppliers be liable for direct, indirect, incidental, or consequential damages resulting from any defect in the software or manual, even if they have been advised of the possibility of such damages. In particular, they shall have no liability for any programs or data stored in or used with Apple products, including the costs of recovering or reproducing these programs or data. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Copyright

This manual and the software (computer programs) described in it are copyrighted by Apple or by Apple's software suppliers, with all rights reserved. Under the copyright laws, this manual or the programs may not be copied, in whole or part, without the written consent of Apple, except in the normal use of the software or to make a backup copy. This exception does not allow copies to be made for others, whether or not sold, but all of the material purchased (with all backup copies) may be sold, given, or lent to another person. Under the law, copying includes translating into another language.

You may use the software on any computer owned by you, but extra copies cannot be made for this purpose. For some products, a multi-use license may be purchased to allow the software to be used on more than one computer owned by the purchaser, including a shared-disk system. (Contact your authorized Apple dealer for information on multi-use licenses.)

Product Revisions

Apple cannot guarantee that you will receive notice of a revision to the software described in the manual, even if you have returned a registration card received with the product. You should periodically check with your authorized Apple dealer.

© Apple Computer, Inc. 1984
20525 Mariani Avenue
Cupertino, California 95014

Apple and the Apple logo are registered trademarks of Apple Computer, Inc.

Simultaneously published in the United States and Canada. All rights reserved.

The Apple II

Offering seven time-tested, entertaining, and educational programs designed to help youngsters as well as adults improve spelling, math, word-recognition, problem-solving and computer-literacy skills; these classic learning games can also be easily customized.

The Apple Education Classics enable you to have fun:

- Building mathematics skills in addition, subtraction, multiplication, and division, as well as in the use of fractions and number lines;
- Discovering information on a wide variety of subjects, ranging from plant and animal life to sports heroes and U.S. history;
- Learning basic economic principles and putting them into practice with Elementary, My Dear Apple's lemonade-stand simulation;
- Designing your own individualized true-false, multiple-choice, and matching quizzes with the help of step-by-step instructions that allow you to grasp BASIC programming concepts.

System Requirements:

- any Apple // system
- one disk drive
- monitor

Recommended:

- color monitor
- two blank floppy disks for backup copies

Apple Computer, Inc., 20525 Mariani Avenue, Cupertino, CA 95014

600-0364-A
A2D4504

The Apple II
The Apple® Education Classics
Elementary, My Dear Apple
For II+, IIe and IIc
Integer Basic
©Apple Computer, Inc. 1980, 1984 691-0008-A

The Apple II

The Apple II
The Apple® Education Classics
The Shell Games
For II+, IIe and IIc
Integer Basic
©Apple Computer, Inc. 1979, 1982, 1984 691-0008-A

The Apple II
apple computer inc.
20525 Mariani Avenue
Cupertino, California 95014
(408) 996-1010
TLX 171-576
914-0312-A

Apple Computer, Inc.

P.O. Box 1143

Cupertino, CA 95014

Tell Apple

About the Apple II

Your Name: _____

Company: _____

Your Title: _____

Address: _____

City/State/Zip: _____

Apple uses comments and suggestions from Apple owners like you to improve existing products and develop new and better products. Now that you've used your Apple II, we want to know your thoughts and suggestions about your experience. Please use this form to Tell Apple what you think.

Your Dealer Can Help: Apple can't respond to your individual questions. If you have a question, would like help, or need service, please contact your Apple dealer.

Part I: Tell Apple About You

1. How much prior experience have you had using computers?

☐ None ☐ Little ☐ Moderate ☐ Extensive

2. Where do you use your Apple computer?

☐ Home ☐ Work ☐ Home and Work

3. Did you own another Apple before buying this one?

☐ Yes ☐ No

4. Which family members use your Apple computer?

☐ Adults ☐ Children

Part II: Tell Apple About Your System

1. Which Apple computer do you have?

☐ Apple II ☐ Apple IIe ☐ Apple IIc ☐ Apple II Plus

2. How much memory do you have? _____

3. Which devices are attached to your system?

☐ Disk Drive ☐ Mouse ☐ Printer ☐ Plotter

☐ Joystick ☐ Hand Controller

☐ Mass Storage ☐ Other(s) _____

(please specify)

4. Which type of display device are you using? (Please check two boxes.)

☐ Video Monitor ☐ Television

☐ Color ☐ Black & White ☐ Monochrome

5. Do you plan to write computer programs?

☐ Yes ☐ No

If yes, which computer language(s) will you use?

_____

6. What application programs do you use or plan to use with your Apple?

☐ Word Processing ☐ Data Management

☐ Education ☐ Games ☐ Communication

☐ Graphics ☐ Spreadsheet

☐ Other: _____

7. Please list products (program, operating system, or language) in use. Please include version number.

Exact Name of Software	Version Number	Product Number

8. Please give us any suggestions or comments and tell us about any problems you've had with your Apple system. Feel free to attach additional sheets. _____

Part III: The Apple Education Classics

Please take a few minutes to tell Apple what you thought of *The Apple Education Classics.*

1. Did the book give you the information you needed to use *The Apple Education Classics*? _____

2. Did you create your own Shell Games? If you did, were the instructions clear? What topics, if any, were not explained thoroughly enough? _____

3. Was the information presented appropriately for your level of computer expertise?

☐ Too Elementary ☐ Too Complex

☐ Just Right

4. Please describe specific problems you encountered in the book and/or programs. (Page numbers would be helpful. Please feel free to attach additional sheets.)

Thank you for your time and effort.

#030-1023-A

The Apple II

The Apple Education Classics

The Shell Games
Elementary, My Dear Apple

Table of Contents

Chapter 1 Introduction **1**

2 A Word About This Manual
2 Making Copies
3 Introducing *Elementary, My Dear Apple*
3 Darts
4 Playing Darts
6 Playing Superdarts
8 Lemonade
9 Supermath
10 Don't Fall
11 Introducing *The Shell Games*

Chapter 2 Modifying The Shell Games **15**

16 Modifying Professor True
18 Modifying Mr. Multiple

Chapter 3 Creating Your Own Shell Games 21

21 Education and Design Philosophy
22 Before You Begin
23 A Quick Overview
24 Anatomy of a Shell Game
24 Getting Inside The Match Machine
26 The Nine Shell Games Principles
29 Creating a New Quiz
33 Entering a New Quiz
35 Using Escape Mode to Edit a Quiz
37 Replacing Text
38 Deleting Text
40 Inserting Text
41 Saving a Quiz

Chapter 4 The Shell Games Editor 43

45 Some Preliminary Steps
48 Typing the Quiz
52 Editing an Existing Quiz
55 Displaying a Quiz
55 Saving a Quiz
55 Searching for a Quiz
58 Verifying the Data Table
59 Replacing an Old Quiz
59 Inserting and Deleting Quizzes
64 Leaving the Editor
65 An Important Fringe Benefit

Chapter 5 Don't Fall Word Lists 67

67 Creating Word Lists
69 Creating Special Word Lists
70 Modifying Don't Fall
71 Editing Word Lists

Chapter 6 Professor True and Mr. Multiple **73**

73 Inside Professor True
73 Some Armchair Philosophy
74 Some Background
75 A Real Example
80 Inside Mr. Multiple
84 General Comments

Chapter 7 Finishing Up **87**

87 Creating New Shell Games Disks
88 Changing Program Names

Appendix Summary **91**

91 General Information
92 Editing The Shell Games
92 Replacing Text
93 Deleting Text
93 Inserting Text
94 The Nine Shell Games Principles
95 The Match Machine
97 Don't Fall Word Lists
98 Professor True
100 Mr. Multiple

Bibliography **105**

106 Note on Lamont Cranston

Index **109**

Introduction

Welcome to *The Apple® Education Classics*, a collection of educational games that are indeed classics: great programs that will endure forever. The games are so varied that there's something for everyone. And you don't have to know a thing about computers to use them.

The Apple Education Classics is a combination of two earlier Apple products: *Elementary, My Dear Apple* and *The Shell Games*. Here's a description of each of the games you'll find:

- **Elementary, My Dear Apple.** A collection of educational games that includes
  - **Darts.** A game that broadens your understanding of fractions.
  - **Lemonade.** A game that lets you turn a lemonade stand into a high-growth enterprise.
  - **Supermath.** Electronic flash cards, complete with explanations.
  - **Don't Fall.** A variation on hangman sure to stump even the sharpest wordsmith.
- **The Shell Games.** A variety of quizzes—some multiple choice, some true/false, some matching—that let you test your knowledge of everything from history to current events to famous sports heros.

  By following the instructions in this manual, you can learn how to create your own Shell Games, and, in the process, learn about the Integer BASIC programming language.

A Word About This Manual

The organization of this manual is based on how you plan to use *The Apple Education Classics*.

- Chapter 1 gives you an overview of the games.
- Chapter 2 tells you how to modify certain rules of The Shell Games. You only need to read this chapter if you want to change the way the games work.
- Chapters 3 through 7 are optional. But if you want to learn how to create your own Shell Games, go through the rest of the book in order.
  - Chapter 3 uses The Match Machine to explain how The Shell Games work.
  - Chapter 4 explains how to use the editor built into The Shell Games. The editor lets you create and modify The Shell Games.
  - Chapter 5 presents a special case: Don't Fall word lists. Although Don't Fall is one of the games on *Elementary, My Dear Apple*, it is designed like The Shell Games.
  - Chapter 6 examines the remaining Shell Games: Professor True and Mr. Multiple.
  - Chapter 7 explains how to rename programs and how to create your own Shell Games disks.
- The appendix is a handy summary of the information presented in this book.

Making Copies

But hold on to your fingers. Before you go ahead and try out the Classics, it's necessary, crucial, and otherwise imperative for you to make backup copies of *The Apple Education Classics* disk. To make backup copies, you can use either the DOS 3.3 COPYA program, the ProDOS Filer on the *ProDOS User's* disk, or the *System Utilities* disk.

To back up *The Apple Education Classics*, you'll need two blank disks:

- One disk for *Elementary, My Dear Apple*.
- One disk for *The Shell Games*.

After you've made your copies, you're ready to try the games. Just put one of your newly made disks in the disk drive (drive 1), turn on your monitor, turn on the computer, sit back, and enjoy. When a menu is displayed, choose whatever option strikes your fancy. The games include instructions.

If All Is Not Well: If, when you start up one of the Classics disks, the picture on your screen looks strange, it may be for one or two reasons. One possibility is that you have an early version of the Apple IIe main logic board (Revision A). To see if you have a Revision A board, take the top off the computer and look at the serial number on the board. If the last letter in the serial number is an *A*, you have a Revision A board. The second possibility is that your Extended 80-Column Text Card is not properly installed. In either case, see your dealer.

Introducing Elementary, My Dear Apple

Elementary, My Dear Apple has four fun-to-play games. The following sections give brief explanations of each.

Darts

Hurry, hurry, step right up,
Test your skill, and try your luck,
Toss the darts, and then you'll see
If you can win a prize from me!

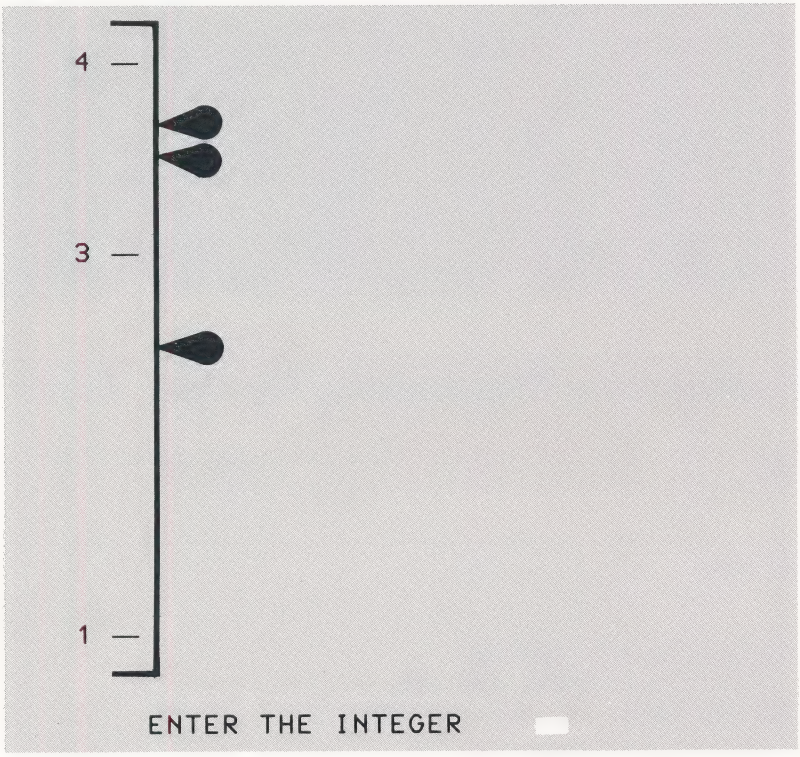
Darts is a game that sharpens your skill with fractions and number lines. The program draws a number line on the screen (a different one each time) and places three balloons on it. The object of the game is to use darts to pop the balloons by

guessing where on the number line the balloons are placed. If you hit all three balloons with four or fewer darts, you're rewarded with a tune.

Playing Darts

After you select Darts from the menu and answer some preliminary questions, you'll see a screen that looks similar to Figure 1-1.

Figure 1-1. *The Darts Display*



Your object is to hit each balloon on the number line with a dart. The program gives you a hint by showing you the numbers at the top and the bottom of the number line, and one number somewhere on the line.

The positions of the balloons are either

- integers (whole numbers, like 4 or 7)
- fractions (like ½ or ⅛)
- or mixed numbers (like 2⅓ or 7⅝).

The numbers are always less than ten, and the fractions will be in halves, thirds, fourths, fifths, sixths, sevenths, eighths, or ninths.

To throw a dart, type the position on the number line where you want the dart to hit. Here's how:

1. Type the integer (whole number) part of the number and then press the SPACE bar. This number should be between zero and nine. If the number you want to type doesn't have an integer part, just press the SPACE bar without typing anything.
2. Type the fractional part. First, type the numerator (the number on the top of the fraction), and press the SPACE bar. Then type the denominator (the number on the bottom of the fraction), and press the SPACE bar again. If the number you want doesn't have a fractional part, just press the SPACE bar instead of typing the numerator.

For example, if your screen shows the number line in Figure 1-1, and you think the middle balloon is at 3½, you would

1. Type 3 (the number).
2. Press the SPACE bar.
3. Type 1 (the numerator).
4. Press the SPACE bar.
5. Type 2 (the denominator).
6. Press the SPACE bar.

The screen shows

```
  1
3 -
  2
```

Now watch the screen—a dart sails across the screen and hits the number line. If the dart hits a balloon, the balloon pops and the exact position of the balloon appears on the left side of the line. When the dart lands, the position of the dart will appear on the right side of the number line.

If you hit all three balloons with four or fewer darts, you win, and the Apple plays a little tune for you (assuming, that is, that you haven't turned off the sound!) Then it draws a new number line, with three more balloons in different places. If you don't want to throw any more darts, just press ESC (on the left side of the computer's keyboard).

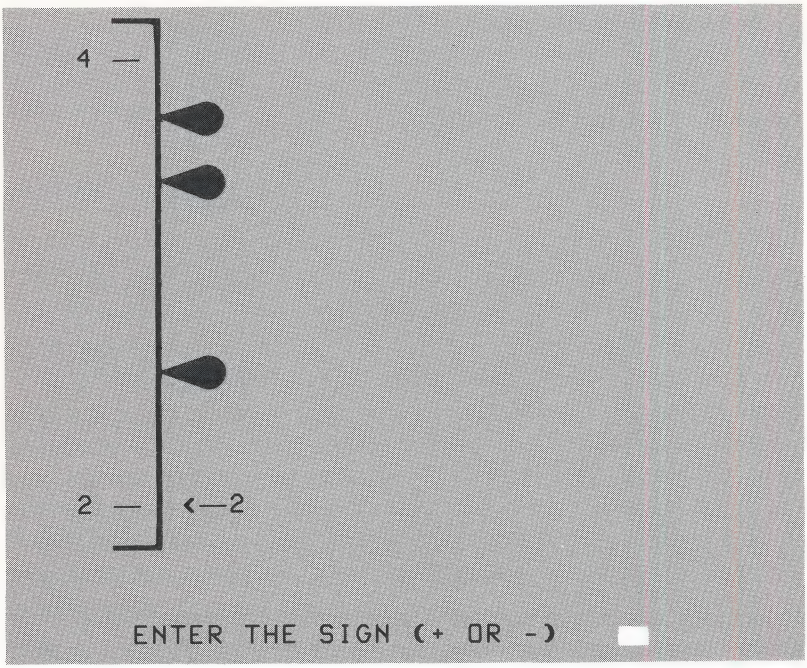
If you make a typing mistake, you can go back and erase the mistake by pressing ←. Each time you press this key, you'll back up one step (from denominator to numerator to integer). Once you've erased the mistake, just type the correct number and continue on as if you had never made the mistake.

Playing Superdarts

In Superdarts, you throw a dart by guessing the difference between the position of the last dart thrown and the balloon. The program always throws the first dart.

Suppose your Superdarts screen looks like Figure 1-2.

Figure 1-2. *The Superdarts Display*



If you think the middle balloon is at 3½, and the last dart landed at 2, then for Superdarts you would

1. Type + (a plus sign).

2. Type 1 (the number).

3. Press the (SPACE) bar.

4. Type 1 (the numerator).

5. Press the (SPACE) bar.

6. Type 2 (the denominator).

7. Press the (SPACE) bar.

The screen shows

```
      1
+1    -
      2
```

In Superdarts, if you want the next dart to land above the last one, make the number positive by typing a plus sign (+) before the integer. If you want to throw the next dart below the last one, make the number negative by typing a minus sign (-) before the integer.

Lemonade

Lemonade is a game that measures your ability to turn your lemonade stand into a profit-making enterprise. The time is midsummer, the place, Lemonsville, California. Each player is in charge of a lemonade stand. Each morning, you check the weather and decide how much lemonade to make for that day. You post signs around the neighborhood and choose a price for your lemonade. Each evening, you tally up your sales and expenses and find out how much money you've made (or lost). Your objective is to make more money than your competitors. The game continues until all players agree to quit.

Everybody starts playing Lemonade with $2.00 in cash and some sugar. Lemonade costs two cents per glass to make; this cost will rise as the days go by. You can make only as many signs and glasses of lemonade as you've got money to pay for. If you don't sell all the lemonade you make, you lose the money you spent making the glasses you didn't sell. Your cash in hand increases as you build your assets.

The amount of lemonade you sell is determined by the price of the lemonade, the weather outside, the number of advertising signs you make, and the number of customers you have. You are not affected by what other players do. And today's performance isn't affected by yesterday's sales.

The weather can change from day to day. The longer you play, the more variety you'll see. The warmer the weather, the more customers you can draw. Sometimes the street department will

be working on your street, which keeps your customers away. On cloudy days, there is a chance of sudden thunderstorms, which ruins everything for that day.

If you decide to quit, just press (ESC) (located on the left side of the keyboard) after the last player's Lemonsville Daily Financial Report. If you like, you can write down each player's assets and the number of days you have played, and continue the game another time.

Supermath

Supermath is an electronic flash card program that not only quizzes you on addition, subtraction, multiplication, and division, but also gives you explanations when you get an answer wrong. Addition and subtraction problems may include numbers up to 999. Multiplication and division problems use numbers up to 12.

In Supermath, the program gives you two new numbers for each problem. The more questions you get right, the harder the next question will be. If you want, you can tell the program to quiz you on a certain number, and it will use that number in every problem. If you can't solve a problem in three tries, the program shows you the answer and explains how to solve the problem step by step.

The program lets you solve the addition and subtraction problems column by column, from right to left. Just type the digits of the answer as you figure them out, and press (RETURN) when you're finished. Many of the harder problems will have you carry or borrow. The result of a subtraction problem is always a positive number.

When quizzing you on multiplication, the program gives you problems up to 12 times 12. When doing division problems, you can ask for problems with or without remainders. The biggest divisor the program gives you is 12. If you ask for problems with remainders, you first type the quotient, then press (RETURN), type the remainder, and press (RETURN) again. Answer multiplication and divison problems from left to right.

If you make a mistake while typing the answer to any problem, you can press ← or → to back up and retype or erase the wrong numbers. When you've typed the answer correctly, press RETURN. When you want to stop, just press ESC.

Don't Fall

Here's a game that's like good ol' hangman, but with a difference. The program chooses a word, and you try to guess it, letter by letter. Each time you guess a letter that is not in the word, you take another step closer to the cliff. If you don't guess the word in time, you fall off the edge!

The program has more than a hundred words in 13 different categories. When the program shows you the list of categories, choose one by typing the number next to the name of the category. Then press RETURN. You also can create your own word lists. Complete instructions for doing this are included in Chapter 5, "Don't Fall Word Lists."

The program then asks you to select a skill level. There are three levels: easy, hard, and expert. The skill level doesn't change how hard the words are, but changes how far away from the cliff you start. For easy questions, you start 13 steps away. For hard questions, you start seven steps away; and for expert questions, you start only four steps away from the cliff!

Next the program asks you whether only humans will be playing or whether the Apple can play too. If you let the program play, then, after you try to guess a word, the program asks you to type a word for the Apple to guess. If you like, the word you type doesn't have to be a word at all; it can simply be a collection of random letters. When the program is trying to guess your word, the robot always stands 13 steps away from the cliff, regardless of the level you have chosen. See if you can fool the program and make the robot fall!

After you pick a category, the program shows you a picture of a person (that's you) walking toward a cliff, and some dashes. There's one dash for every letter in the word. Guess a letter that you think is in the word, type it, and press RETURN. If your

letter is in the word, the program puts the letter in place over the dash. If your letter isn't in the word, you take another step closer to the cliff! Too many wrong letters, and you fall! *Ouch!*

Introducing The Shell Games

The Shell Games are so named because the games are *shells* that you can use to create your own quizzes on any subject. The program that asks the questions, gives instructions, and keeps track of answers is already written. You can easily modify these programs, using a special editor, to contain questions and answers on any subject you like.

There are three Shell Games: The Match Machine, Professor True, and Mr. Multiple.

- **The Match Machine.** A collection of 15 matching quizzes, from sports superstars to secret identities. Watch the screen as the choices get shuffled before your eyes.
- **Mr. Multiple.** A grabbag of multiple choice stumpers that run the gamut from who *really* wrote the Autobiography of Malcolm X to the Buster Brown show line-up. Learn the answers and then impress your friends.
- **Professor True.** A potpourri of true/false querries about America's past, starting with Columbus. Full explanations are included.

There are three different ways to use The Shell Games:

- You can play The Shell Games that come on the disk, matching your wits against the computer's.
- You can modify the rules of the existing games, changing things like the amount of encouragement users are given during the course of a quiz. Chapter 2 explains how to make minor modifications to The Shell Games.
- You can create your own games. To do this, it helps to have some knowledge of BASIC, although, it's certainly possible to modify The Shell Games without knowing beans about BASIC. Instructions for creating Shell Games start in Chapter 3.

The Shell Games are easy to use. Just follow the instructions on the screen and have a good time. So, pop in your copy of *The Shell Games* disk and give the games a whirl.

Modifying The Shell Games

Both Professor True and Mr. Multiple have options that let you change the way the games run. These options are often called *flags*. That's because flags are usually found in one of two ways: raised or lowered. Similarly, options can be either on or off. Here's how to look at the flags that control how the games work:

1. Select the game you want to modify (either Professor True or Mr. Multiple) from The Shell Games Menu.
2. First reply to the question asking if you want sound. Then, when you see the message asking you to type your first name, press CONTROL-C; then press RETURN. (To press CONTROL-C, hold down CONTROL, then press C, then release both keys.)
3. Next, type

```
LIST 9000, 9009
```

One way to change a flag is to retype the line that contains the flag. Another, more efficient method, is to use escape mode to copy over everything but the yes/no change you want to make. You can learn how to use escape mode in Chapter 3, "Using Escape Mode to Edit a Quiz."

Modifying Professor True

The rules for Professor True and Mr. Multiple may be modified to fit your needs and educational philosophies. This section tells you how to modify the programs that come on *The Shell Games* disk. Later you'll learn how to create your own quizzes.

To modify Professor True, select PROFESSOR TRUE from the menu, press (CONTROL)-(C) and then (RETURN). Then type

```
LIST 9000, 9009
```

to see how the program works. First, you see four string variables, named D1$ through D4$. These strings determine the content of your quiz's title page. Each string holds one line (38 characters) of information, and each line will be automatically centered on the title page screen. You need use only as many lines as you wish.

Following these strings, on line 9003, is the only Professor True feature that is not also found in Mr. Multiple: PROFSPEAKS. PROFSPEAKS sets the frequency with which Professor True will give the user encouragement. It may be set to any number from zero to ten—zero is never, ten is on each question. You may select the number based on the age group you are trying to reach, your own philosophies, or the amount of encouragement you have written into the questions themselves. And don't forget that the Professor tends to give out extra points rather freely. So, the higher the setting, the more points the user may get.

When you modify the flags, keep in mind that they can be changed at any time. The first time you have people take the quiz, the information will be new and you will want them to read all of it. A few weeks or months later, you may want the same people to take the same quiz, but this time skip the explanation if they answer correctly, or you may want to use one of the other options. Just change the answers to the flags from NO to YES, or YES to NO, and you have an entirely new format for the same quiz.

Naming Variables: *EXPLANATION* and *SCORING* are mispelled when they are part of the names of flags to avoid the reserved words, *AT* and *OR*, contained in the properly spelled versions. (A variable name—or flag name—cannot be or contain a BASIC reserved word.)

Here is a list of the flags and what they do:

- SKIPEXPLANACIONIFRIGHTANSWERFLAG=YES causes the program to give the user more information only if the user misses the question. This can be useful when the material in the quiz is not new to the user and the explanation is essentially a review. You may wish to allow users who know the subject matter to breeze through the quiz, leaving more time for users whose knowledge is a little shakier. This flag has no effect on the second round of the quiz. When the user retakes the missed questions, the program will always skip the explanation if the question is answered correctly.
- ALLOWEXPLANACIONSKIPFLAG=YES allows the user to prematurely press the (SPACE) bar and skip the explanation before it has been completely displayed.
- SKIPEXPLANCIONFLAG=YES causes the program to skip all explanations. There is no option given to the user; so as far as the user is concerned, the quiz doesn't include explanations.
- SKIPSCØRINGFLAG=NO eliminates scoring and makes PROFSPEAKS equal to zero.
- MUSTANSWERALLQUESTIONSFLAG=YES eliminates the option of using (ESC) to skip the balance of the questions. This forces users to answer every question before they can exit from the program. All questions do not have to be answered correctly, however.
- MUSTRETESTWRONGANSWERFLAG=YES causes the user to go back and answer all missed questions before the user can exit the quiz. It does not force the user to answer all the questions the first time through.

The combination of MUSTANSWERALLQUESTIONSFLAG=YES and MUSTRETESTWRONGANSWERFLAG=YES will create a quiz in which each question must be answered correctly before the game ends.

By setting these flags to YES or NO, you can customize the quiz to suit your own purposes. For example, if you want to be sure that your users have indeed learned the answers to a quiz they have been allowed to take at their leisure, you might set the flags to require their answering every question and getting each one right. You might not set the SKIPEXPLANACIONSFLAG; you might or might not allow users to voluntarily skip the explanations with the ALLOWEXPLANACIONSSKIPFLAG set to YES.

As with any other change you make to The Shell Games, you must save the version with changed flag settings on a disk if you want to use that version in the future. If, for example, you change the flags in Professor True, you would save the new version by typing

```
SAVE PROFESSOR TRUE
```

The new flag settings won't be in effect until you run the revised version.

Modifying Mr. Multiple

Mr. Multiple has the same title page strings and the same flags as Professor True, except that there is no PROFSPEAKS in Mr. Multiple. To see Mr. Multiple's flags, type

```
LIST 9000, 9009
```

Then follow the instructions for modifying Professor True in the previous section.

Creating Your Own Shell Games

The Apple makes learning fun; it should make teaching just as interesting. Designing a well thought-out quiz is difficult and creative work. This book should make it easy for you to create your own quizzes, as well as edit or add to those already created. The rest of this book tells you everything you need to know to make and modify your own quizzes.

Don't Take Notes: The appendix is a handy summary of the information in this book.

Education and Design Philosophy

Before delving into the secret world of the quiz editor (called *The Shell Games Editor*), here are a few words about the educational philosophy of The Shell Games. As you use these games, notice that they include a maximum amount of positive reinforcement and an absolute minimum of negative sanction. Because the quizzes were designed primarily to teach, not test, there is no penalty for missing a question. But the user is encouraged to do the missed questions over again until they are answered correctly, thereby receiving the color and sound reward.

The backward problem format of Professor True and Mr. Multiple, with the explanation following the question, has proved to be very effective. With new information, the user often will miss the first time, but an intriguing question will raise interest in the answer that follows. The user's comprehension

and retention the second time around has proved to be high. And the same users a week or two later have continued to show high retention of information. This has proved to be true even for young children. Today, The Shell Games are being used in a wide variety of classroom settings, ranging from special education to premed, as well as in thousands of homes.

If you are a teacher, you will soon discover two methods of making new quizzes. One is doing the entry yourself. The other is by having the students do the entry! This has several advantages, not the least of which is saving your time. Beyond that, however, you give your students the chance to develop their computer literacy by doing useful programming that they will be able to point to with pride. And they will not only learn more about computers, but will automatically learn the contents of the quiz. Try it and see. It works!

The Shell Games can be used by young students as well as adults so it is important that they be as easy to use as possible. For this reason, the material for The Shell Games (your quizzes) becomes part of the program itself, so a user doesn't notice any change.

Before You Begin

The Shell Games are written in Apple Integer BASIC, a dialect of BASIC similar to Applesoft BASIC, the variety of BASIC most often used on the Apple II family of computers. If you are familiar with Applesoft BASIC, the main difference you will notice between it and Integer BASIC is the prompt. The Applesoft BASIC prompt looks like this:

```
]
```

The Integer BASIC prompt looks like this:

```
>
```

BASIC: BASIC is a computer language patterned after English, which makes it easy to use. BASIC stands for *Beginners All-purpose Symbolic Instruction Code*.

While there is no need for you to have an extensive knowledge of BASIC to create new quizzes, some understanding of the BASIC language will make creating the quizzes easier. However, all the instructions for creating quizzes are presented in an easy-to-follow, step-by-step form. So, even if you don't know BASIC, you should have no difficulty making new quizzes. Still, don't be concerned if you don't understand all the program's intricacies.

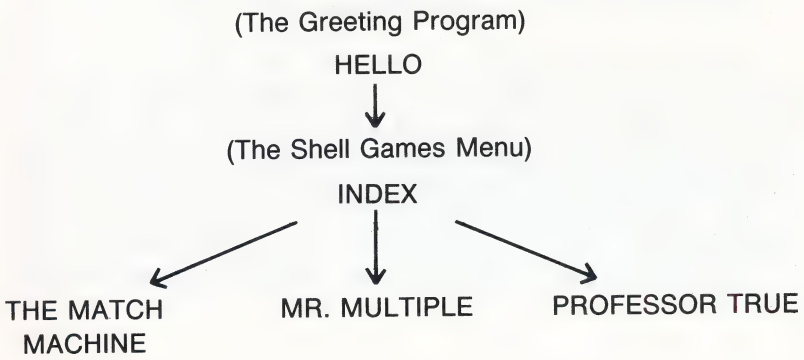
When you are ready to create quizzes, you will use The Shell Games Editor, Chapter 4. But first, an overview. This chapter will concentrate on the basics. As you read on, additional features and powers will be pointed out. So, go through the remainder of the book in order. And try each step on your computer.

A Word to the Wise: It's easy to learn to create Shell Games of your own if you follow the instructions in order. Otherwise, things can get a bit confusing. So, read this book from the beginning.

A Quick Overview

Figure 3-1 shows how the programs on *The Shell Games* disk are arranged.

Figure 3-1. *Organization of The Shell Games*



When you start up *The Shell Games* disk, the program named HELLO immediately runs INDEX, the program that creates the main menu for The Shell Games.

From the menu, you may select any of the three Shell Games.

When you are using a Shell Game, look for instructions on the screen. (You can usually press ESC to exit from a Shell Game.)

You can stop a program in progress by pressing CONTROL-C and then RETURN. When you do, you see the Integer BASIC prompt. To return to The Shell Games Menu, type

```
RUN INDEX
```

Warning

When you type RUN INDEX*, the program you have been using disappears from memory. As you will see, it is not always desirable to eliminate the current game from memory. If you modify a game and want to keep the changes you've made, you must save the game before running INDEX.*

Anatomy of a Shell Game

This section uses The Match Machine to explain how The Shell Games are designed. Once you know how The Match Machine works, understanding the other games is easy.

Getting Inside The Match Machine

Before you can create a new Match Machine quiz, you need to see what an existing quiz looks like. Here's how:

1. Put The Match Machine program from the disk into the computer's memory by selecting option 1, THE MATCH MACHINE, from The Shell Games Menu.

2. After you tell the program whether you want sound, you see a title page. The last line of the screen shows:

```
PRESS THE SPACE BAR TO BEGIN
```

Instead of pressing the SPACE bar, press CONTROL-C. (While holding down CONTROL, press C, then release both keys.) This stops The Match Machine program. Then press RETURN. You see the Integer BASIC prompt:

```
>
```

3. To look at the first Match Machine quiz, type

```
LIST 9010, 9014
```

If no one has modified your Shell Games disk, you will see:

```
9010 D1$="ANIMALS AND THEIR YOUNG,ANI
     MAL&ITS YOUNG,BEAR&CUB,BIRD&NEST
     LING,CAT&KITTEN,COW&CALF,DOG&PUP
     PY,FISH&FRY"
9011 D2$="FROG&TADPOLE,GOAT&KID,KANGA
     ROO&JOEY,SHEEP&LAMB,BUNNY,KIT,EL
     VER": RETURN
9012 REM
9013 REM
9014 REM
```

These are the lines of The Match Machine program that produce the matching quiz Animals and Their Young. Once you understand these five lines and why they look as they do, you will have grasped 95 percent of what you need to know to write your own quizzes. So those of you who are new to programming—*don't panic!*

Confused? It doesn't matter if your screen shows a different quiz. The same principles will apply.

Notice that all examples in this manual appear in uppercase. If your computer has uppercase/lowercase capabilities, you can type the quiz questions in uppercase and lowercase. The actual BASIC commands must be typed in uppercase.

Apple II and II Plus Users: If you see a lot of numbers and symbols where lowercase letters should be, it is because your computer can't display lowercase. When you play The Shell Games, the quizzes are automatically changed to uppercase, but when you list them, as you are doing now, they will look funny.

If you want to see the real letters (in uppercase), copy over the lines, as explained in the chapter on editing. Every character you pass the cursor over, by pressing →, will magically be transformed into uppercase. For now, just look at the listing in the book.

The Nine Shell Games Principles

All Shell Games, whether they appear in The Match Machine, Mr. Multiple, or Professor True, share the same nine characteristics. Here's a list of those characteristics.

1. The first quiz in all Shell Games always begins on line 9010.
2. All quizzes occupy exactly five consecutive program lines: 9010 to 9014 for the first quiz, 9015 to 9019 for the second quiz, 9020 to 9024 for the third quiz, and so on. All of the program lines containing quizzes make up an area of the program called the *data table*.

Screen Lines Versus Program Lines: A BASIC program line is not the same as a single line of text on your screen. A program line often is composed of several lines of text. When you press RETURN, the computer knows that a program line is complete.

3. Every Shell Game includes a maximum of three kinds of BASIC statements: string statements, RETURN statements, and REM statements. Look at the Animals and Their Young quiz; then read the following explanation.

   String Statements. In the world of BASIC, a string is a bunch or string of characters surrounded by quotation marks. The string statements (in this case, lines 9010 and 9011) determine the content of the quiz. Each string statement begins with the name of the string (in this case, D1$ and D2$) followed by an equal sign (=), followed by the content of the string in quotation marks.

   RETURN Statements. A RETURN statement contains the word *RETURN*. Whenever BASIC sees RETURN, it knows to return to a specified part of the program.

   REM Statements. A REM statement acts as a place holder. In The Shell Games, a REM statement always occupies a line of its own.

4. Each quiz can have up to five strings. In The Shell Games, the strings are always named D1$, D2$, D3$, D4$, and D5$.

5. Every quiz must begin with D1$. That's because the program starts a quiz each time it finds a D1$. If you were to write a quiz that began with F7$ or QUIZ$, the program would ignore it. The program knows it has run out of quizzes when it fails to find a D1$.

6. While you only use as many of the five strings as you need, the strings must be in ascending order. For example, you cannot write a quiz in which D4$ is followed by D3$. It is not necessary to use each string in order. You may, for example, use D1$, D2$, and D4$ for a given quiz without using D3$ at all.

Laying the Groundwork: Information you place inside the strings will be discussed after you complete an exploration of the external structure of the program lines. Please keep in mind that this section is discussing things that all The Shell Games have in common.

7. A RETURN statement always follows the last string used. In this example, the RETURN statement appears on line 9011. Notice that RETURN comes after the quotation mark and is separated from the string statement by a colon. You also could put the RETURN statement on a separate line, like this:

```
9012 RETURN
```

RETURN also could be on line 9013 or 9014—anywhere will do as long as the RETURN statement appears after the last string.

8. A quiz must use all five line numbers. So, after you type your quiz into the strings and have put in a RETURN statement, check to see if you have any line numbers remaining. If you do, type each of those line numbers, and then type REM, just as in lines 9012 through 9014 in the previous example.

(In the world of BASIC, REM is short for *remark.* Usually, the REM statement is used in BASIC programs so that the programmer can include remarks in the body of the program. In this instance, the REM statement is used as a place holder.) It is not necessary to type any spaces or characters after REM.

9. The final quiz is not a quiz at all. It is a dummy RETURN statement. It is called the *data table terminator* (since it terminates the data table). The data table terminator must

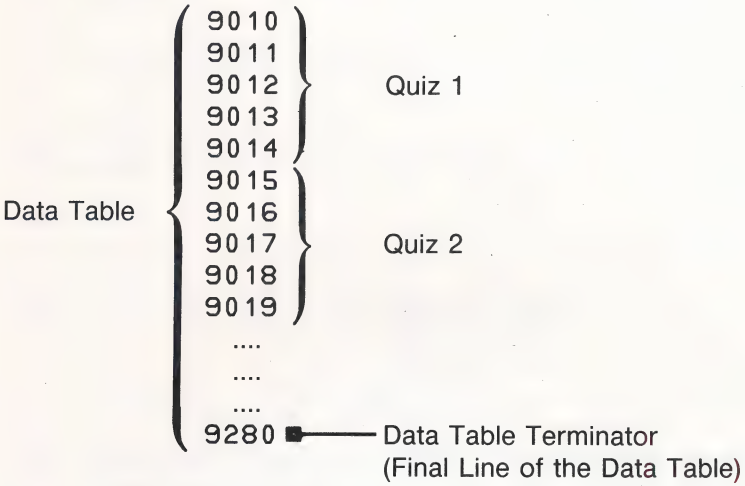
- have a line number that is a multiple of five
- be placed where the first line of the next question or quiz would normally start
- be a simple statement, with RETURN on a line by itself.

For example:

```
9085 RETURN
```

Figure 3-2 should give you some idea of the way The Shell Games are arranged.

Figure 3-2. *Shell Game Anatomy*



This completes the discussion of characteristics that apply to all The Shell Games. The next section shows you how to add a new quiz to The Match Machine.

Creating a New Quiz

You can create Match Machine quizzes with or without the help of The Shell Games Editor. This section shows you how to create quizzes without the editor. Using the editor is explained in Chapter 4. After you've tried both methods, you can decide which you like best.

Creating Other Quizzes: This section tells you how to create a Match Machine quiz. Instructions for creating Professor True and Mr. Multiple quizzes are in Chapter 6.

When you create a new Match Machine quiz, you supply as many as four different sets of information for each quiz you create.

- The quiz title that you want to appear on the menu. This is mandatory.
- The headings for the two columns to be matched, tied together with an ampersand (&). This is mandatory.
- The matches themselves, each tied to its answer by an ampersand, each group separated by commas. The number of matches is optional.
- Extraneous answers. These are optional.

Take a closer look at these four parts. If you look at the listing on your screen or at the sample Animals and Their Young listing, first you will see the title:

```
ANIMALS AND THEIR YOUNG
```

This title will appear on the list of categories that is displayed after you choose THE MATCH MACHINE from The Shell Games Menu. The title is followed by the column headings

```
ANIMAL&ITS YOUNG
```

and the individual matches

```
BEAR&CUB,BIRD&NESTLING,
```

and so on.

Note that each match is connected by an ampersand (&) and each set or title ends with a comma (,). The Match Machine will correctly link up duplicate matches. If you have, for example, BEAR&CUB and TIGER&CUB, the program will not care which CUB is matched with which animal. After the matches are some extraneous names of baby animals. These make the right column longer than the left. The extraneous names are

```
BUNNY,KIT,ELVER
```

You should think of the five strings, D1$ through D5$, as one continuous string of characters, which happens to be broken into five easy pieces. When you select a quiz, the program ties the strings together again into one piece. That's why it doesn't matter how the strings are divided. The only rule you must

observe when dividing up your quiz among strings is that breaks between strings must occur between sets of matches, where you would normally place a comma. This way the individual matches, such as FISH&FRY, FRUIT&APPLE,ONE&UNO, and so on, are not split apart.

These lines would confuse the program since they split up the match.

```
9011 D2$="ALLIGATOR"
9012 D3$="&PEAR,CROCODILE&TEARS"
```

The program does not care where or even whether you include the comma at the end of a match. In the following examples, you can see that the comma may be at the end of the first string, at the beginning of the second string, or eliminated completely.

```
9011 D2$="ALLIGATOR&PEAR,"
9012 D3$="CROCODILE&TEARS"
```

```
9011 D2$="ALLIGATOR&PEAR"
9012 D3$=",CROCODILE&TEARS"
```

```
9011 D2$="ALLIGATOR&PEAR"
9012 D3$="CROCODILE&TEARS"
```

A Word to the Wise: All spaces count in The Match Machine! Unless you want to see a space preceding a word on the screen, don't put a space in the quiz.

Here is a summary of the information that makes up each Match Machine quiz.

Element	**Maximum**
1 Quiz Title	34 Characters
1 Pair Column Headings	15 Characters Left Column Heading
	18 Characters Right Column Heading
Up to 10 Pairs of Matches	15 Characters Each Left Column Entry
	18 Characters Each Right Column Entry
4 Extraneous Answers (Optional)	18 Characters Each Entry

The following special symbols are used:

Ampersand (&)	Connects the Matches
Comma (,)	Marks the End of a Match Set (or Title)

Note that you can have as many as ten sets of matches and that the extraneous answers are optional.

Entering a New Quiz

You are now ready to enter a new quiz. The quiz you will create will be called Elementary Math and will look like this:

PROBLEM	ANSWER
3 + 4	7
15 - 6	9
21 / 3	7
2 x 4	32 / 4
	4
	17 + 5

Note that the matches are 3 + 4 and 7, 15 - 6 and 9, 21 / 3 and 7, 2 x 4 and 32 / 4. The extraneous answers are 4 and 17 + 5.

When typing this quiz, do not press (RETURN) until you are through typing an entire string. The program automatically starts putting text on the next line when you run out of space at the end of a line on the screen.

If, when typing the program lines, you notice that you've made a mistake before you've pressed (RETURN), you can fix your mistake (using the arrow keys) without having to retype the whole line. Here's how:

1. Press (←) until the cursor is over the mistake. Each time you press (←), the character the cursor is sitting on is deleted, even though it still shows up on the screen.
2. Type the correction.
3. If the rest of the line is correct, press (→) to copy over the remainder of the line. Otherwise, retype the remaining portion of the line.
4. Press (RETURN).

If you notice that you've made a mistake after you press (RETURN), you can type the line over again or you can edit the line. Instructions for replacing, inserting, and deleting characters

without retyping the entire line are given in the next section, "Using Escape Mode to Edit a Quiz." The Shell Games Editor is explained in Chapter 4.

To enter the Elementary Math quiz, follow these steps:

1. Clear out the quiz that currently is stored on lines 9010 to 9014 by typing:

   ```
   DEL 9010, 9014
   ```

   (Remember, you won't lose the original version of any Shell Game unless you purposefully save the new version on top of the old. So while you are modifying the version in the computer's memory, the original remains safely undisturbed on your backup disk.)

2. Type line 9010 of the quiz. It should look like this:

   ```
   9010 D1$="ELEMENTARY MATH,PROBLEM&ANSWE
   R,3 + 4&7,15 - 6&9,21 / 3&7"
   ```

   You could include more matches in this string, but there are four more empty strings available, so it is not necessary to crowd them. Also, the spaces between the symbols and the numbers are optional. Remember that if your computer has lowercase letters, you can type the quiz—that is, everything between quotation marks—in uppercase and lowercase. BASIC commands must be all uppercase.

3. Type line 9011 so it looks like this:

   ```
   9011 D2$="2 x 4&32 / 4"
   ```

4. It is now time to finish up with the extraneous answers. These could easily follow upon the heels of 32 / 4 in D2$, but use line 9012:

   ```
   9012 D3$="4,17 + 5"
   ```

5. Now, you need to type a RETURN statement that includes a line number and the word *RETURN*. Type

   ```
   9013 RETURN
   ```

Note: Don't put variable names, like D1$ and D2$, in RETURN or REM statements.

6. You now need to fill line 9014. Because you already included the mandatory RETURN statement in the quiz, just put a REM statement in line 9014. Type

```
9014 REM
```

To see your quiz in action, type

```
RUN
```

You will see The Match Machine Menu with a new category at the top:

```
1 ELEMENTARY MATH
```

If you select 1, you will see your new quiz. If all does not appear well, exit from the program by pressing (CONTROL)-(C); then list lines 9010 to 9014 by typing

```
LIST 9010, 9014
```

and compare them with the instructions. If you find an error, you'll need to edit your program. The next section explains how to edit The Shell Games programs you create. If you're not familiar with editing Integer BASIC programs, you probably should look over this section, even if your quiz doesn't contain any mistakes.

Using Escape Mode to Edit a Quiz

You've already learned about using (←) and (→) to correct mistakes that you've made before you reach the end of a program line. Most often, however, you won't notice a mistake until after you have pressed (RETURN), and the program line is tucked away in the computer's memory. To correct a program line like this, you need to use *escape mode*. Escape mode lets you move the cursor without affecting what's on the screen or in the computer's memory.

To get into escape mode, press (ESC). To leave escape mode, press the (SPACE) bar. Once you are out of escape mode, all you have to do to get back in is press (ESC).

Once you are in escape mode, here's how you move the cursor:

Press (I) or (↑)	To move the cursor up.
Press (M) or (↓)	To move the cursor down.
Press (J) or (←)	To move the cursor left.
Press (K) or (→)	To move the cursor right.

More About Editing: If you are using an Apple computer that has four arrow keys ((→), (←), (↑), and (↓)), you can use either the arrow keys or (I), (J), (K), and (M) to move the cursor in escape mode. If you are using a computer that has only (→) and (←), you must use (I), (J), (K), and (M) for editing.

The following sections explain how to edit text using (I), (J), (K), and (M). This editing method is emphasized because it's more straight-forward than using the arrow keys. When you enter escape mode and then use the arrow keys, there's no immediate feedback as to whether you're in or out of escape mode. When you use (I), (J), (K), and (M) as editing keys, you can immediately tell if you've exited from escape mode. If you aren't in escape mode, the character of the key you press (I, J, K, or M) will be displayed.

Here is some additional information about escape mode:

- To correct or change your text, use escape mode to position the cursor over the first digit in the line number of the line you want to fix.
- Escape mode allows you to correct only one program line at a time.
- After you make a change in a line, copy over the remainder of the line using (→).
- With certain versions of the Apple II, when you are in escape mode, the cursor will change to a plus sign (+) that is displayed in inverse (dark on light).

Apple II Owners: If you own the earliest member of the Apple family, the Apple II, you will find that (I), (J), (K), and (M) do not move the cursor when you are in escape mode. To move the cursor around the screen, you will have to use these key sequences:

Press (ESC)-(A)	To move the cursor one space to the right and exit from escape mode.
Press (ESC)-(B)	To move the cursor one space to the left and exit from escape mode.
Press (ESC)-(C)	To move the cursor down one row and exit from escape mode.
Press (ESC)-(D)	To move the cursor up one row and exit from escape mode.

Note that you must press (ESC) before each press of an editing key. Also, you don't need to press the (SPACE) bar to exit from escape mode. The Apple automatically exits from escape mode after you press an editing key one time.

Replacing Text

This example demonstrates how to use (I), (J), (K), and (M) in escape mode to replace incorrect characters. Specifically, you'll change line 9012 of the Elementary Math quiz so that the first extraneous answer is 8 instead of 4.

1. List the program line you want to modify by typing `LIST 9012`. You should see this line:

```
9012 D3$="4,17 + 5"
```

If you didn't try the Elementary Math quiz, type line 9012 now.

Bailing Out: If you get confused when trying to edit a program line, or if you see

```
***SYNTAX ERR
```

(a syntax error message) on your screen, you can start over any time by retyping the line. To exit from a line you are in the middle of typing, press (CONTROL)-(X). If you are in escape mode, you must exit by pressing the (SPACE) bar before you can press (CONTROL)-(X).

2. Next you need to move the cursor to the first character of line 9012. First enter escape mode by pressing (ESC); then press (I) until the cursor is on the 9, the first character of line 9012.

3. Now that the cursor is where you want it, exit from escape mode by pressing the (SPACE) bar. Notice that the cursor doesn't move the first time the (SPACE) bar is pressed.

4. Press (→) ten times, or until the cursor is on 4. Even though the complete program line shows on the screen, the program line in memory is now

   ```
   9012 D3$="
   ```

   Notice that you always use (→) to copy over the portion of the line you want to keep intact.

5. Next, type 8. What you are doing here is replacing the old character with the new.

6. Press (→) to copy over the remainder of the line. If you don't do this, everything after 8 is erased.

7. Press (RETURN).

8. List the new line 9012 by typing `LIST 9012`. Your new line should look like this:

   ```
   9012 D3$="8,17 + 5"
   ```

Deleting Text

When editing program lines, you'll frequently want to delete extraneous characters. To delete characters, you use escape mode to pass over the characters that you want to delete.

In this example, you'll delete the first extraneous answer in line 9012, so that the line looks like this:

```
9012 D3$="17 + 5"
```

To make the deletion, follow these steps:

1. List line 9012 by typing `LIST 9012`. The line should look like this:

   ```
   9012 D3$="8,17 + 5"
   ```

   If your line 9012 doesn't look like this, type it correctly now.

2. Enter escape mode by pressing (ESC); then press (I) to move the cursor so it's on 9, the first character of line 9012.

3. Once you have the cursor where you want it, exit from escape mode by pressing the (SPACE) bar.

4. Use (→) to copy over the line up to 8, the first character you want to delete.

5. Enter escape mode by pressing (ESC). Then, press (K) twice to move the cursor over the characters you don't want to copy. By not copying these characters, you are deleting them. Your screen will look like this:

   ```
   9012 D3$="8,17 + 5"
   ```

   Note: Even though the characters you want to delete still show on the screen, they will be deleted from the program line in memory. To copy characters into the computer's memory, you have to use (→).

6. Exit from escape mode by pressing the (SPACE) bar.

7. Press (→) to copy over the remainder of the line and press (RETURN).

8. Type `LIST 9012` to check out your handiwork. Your new line should look like this:

   ```
   9012 D3$="17 + 5"
   ```

Inserting Text

Inserting characters also requires moving in and out of escape mode. In this example, you'll put line 9012 back in its original form, so it looks like this:

```
9012 D3$="4,17 + 5"
```

To do the insertion, follow these steps:

1. List line 9012. You should see this line:

   ```
   9012 D3$="17 + 5"
   ```

2. Enter escape mode and press (I) to move the cursor to the beginning of line 9012. Then, exit from escape mode.

3. Press (→) ten times or until the cursor is on 1 in 17. The screen looks like this:

   ```
   9012 D3$="17 + 5"
   ```

   But the program line in memory looks like this:

   ```
   9012 D3$="
   ```

   because you have copied the line only up to the cursor.

4. Enter escape mode and press (J) twice. This creates two spaces for the new text you want to enter. Remember, when you move the cursor while in escape mode, you don't change the program line inside the computer's memory. It still reads

   ```
   9012 D3$="
   ```

5. Press the (SPACE) bar to leave escape mode.

6. You now can insert your new text. Type

   ```
   4,
   ```

   On the screen, line 9012 looks like this:

   ```
   9012 D3$4,17 + 5"
   ```

   but in memory, believe it or not, it looks like this:

   ```
   9012 D3$="4,
   ```

7. To copy over the remainder of the line, press → until the cursor is past the final quotation mark; then press RETURN.

8. List the new line. It should look like this:

```
9012 D3$="4,17 + 5"
```

All of this slight-of-hand may seem strange at first, but after you do it a few times, you'll get the hang of it.

Saving a Quiz

After you have corrected your quiz, type RUN to see if the revised version works properly.

If you want to save this quiz, now is the time to do it. But you should know that if you save this quiz, you'll lose the first Match Machine quiz included on your backup copy of *The Shell Games* disk. Later you'll learn how to add a new quiz to the collection of Match Machine quizzes without losing any of the old quizzes.

If you want to save the version of The Match Machine with the Elementary Math quiz (but lacking the Animals and Their Young quiz), type

```
SAVE THE MATCH MACHINE
```

when the BASIC prompt is displayed. This replaces the previous version of The Match Machine on your disk.

The Shell Games disk only has room for a few new quizzes. You can, however, create a new disk that contains a different set of Shell Games. Instructions for doing this are in Chapter 7.

The Shell Games Editor

The quiz entry scheme explained in the last chapter works very well when you want to delete one whole series of quizzes and replace it with a new set. But what if you just want to add one quiz onto the end of the existing set? First, you have to find the end of the existing data table. Then you have to make sure you create a new data table terminator line (Shell Games Principle 9). To make this process easier, your disk includes The Shell Games Editor.

The Shell Games Editor lets you design your own quizzes. Each Shell Game includes its own editor, and each editor is identical except for some wording differences. In this chapter, you will learn how to create a quiz, called Animal Classification, using this built-in editor.

Here's how to display The Shell Games Editor:

1. Select `THE MATCH MACHINE` from The Shell Games Menu.
2. When you see The Match Machine title screen, press (CONTROL)-(C). Pressing (CONTROL)-(C) stops The Match Machine program.
3. Next, type

   ```
   RUN 2
   ```

   This runs The Shell Games Editor.

Your screen should look like Figure 4-1.

Figure 4-1. *Shell Games Editor Menu*

```
        *** SHELL GAMES EDITOR ***

PLEASE CHOOSE FROM MENU:

1> ENTER A NEW QUIZ

EDIT:

2> EDIT CURRENT QUIZ: #1

3> DISPLAY CURRENT QUIZ: #1

4> SEARCH FOR QUIZ TO BE EDITED

5> VERIFY DATA TABLE FORMAT

6> VERIFY TABLE FORMAT TO ALLOW RENUMBER

7> END

PLEASE ENTER THE NUMBER OF YOUR CHOICE
AND PRESS <RETURN>
```

This chapter will take you through the steps of creating a new quiz with The Shell Games Editor. First, you'll do some preliminary housekeeping chores. Then you'll start entering the actual quiz.

Some Preliminary Steps

Before you can add another quiz, you need to let The Match Machine program know that a new quiz is coming so that it can make room for it. To do this, follow these steps:

1. Select 1, ENTER A NEW QUIZ, from The Shell Games Editor Menu.

   The first thing that happens is that the editor scans through the first lines of the existing quizzes, looking for the data table terminator line: the first line of a quiz (an even multiple of five) containing only a RETURN statement.

   When the editor finishes, your screen will look similar to Figure 4-2.

Figure 4-2. *Finding Lines for a New Quiz*

```
==== FINDING LINES FOR NEW QUIZ ====


PLEASE COPY OVER THE FOLLOWING LINES,
USING THE RIGHT ARROW, AND PRESSING
THE <RETURN> KEY AT THE END OF EACH:

>9090 RETURN <== PRESS <RETURN> HERE
GOTO 4340 <== PRESS <RETURN> HERE
```

Screen Look Different? If your screen doesn't look like Figure 4-2, it's because your program may have a data table terminator statement (that is, a RETURN statement) in a different place. People tend to leave data table terminators lying around rather casually. Your program doesn't mind—as long as there is one. So, if your screen looks like Figure 4-3, skip to step 3.

If the line numbers on your screen do not correspond to the ones in Figure 4-2, you have an altered version of the program. The specific line numbers you see are unimportant.

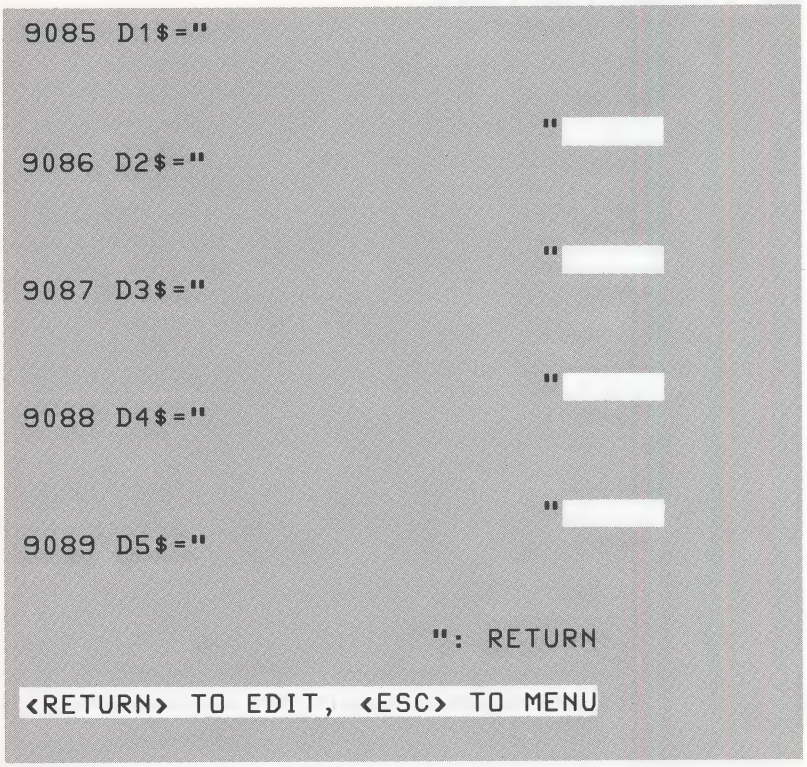
2. The next step is to copy over the BASIC program lines. If you look at your screen, you'll see that the cursor is positioned at the beginning of the first line. Just press → to step across each of the characters, until you reach the space just before the arrow displayed in inverse (dark text on a light background).

Then press RETURN. This will position the cursor at the beginning of the next line. Step across the second line by pressing → and again press RETURN just before the arrow that is in inverse.

Really, when you copy over these lines, you are creating a new data table terminator line five lines ahead of the existing data table terminator line. This makes room for the new quiz you are creating. As mentioned earlier, every Shell Game must end with a data table terminator line: a dummy RETURN statement on a line number that would normally be the first line of the next quiz.

When you're finished copying the lines, your screen should look like Figure 4-3.

Figure 4-3. *Creating a New Quiz*



3. The last line of text on the screen asks you whether you want to start editing or return to The Shell Games Editor Menu:

    <RETURN> TO EDIT, <ESC> TO MENU

    If you had embarked on this course accidentally, you could press (ESC) to return to The Shell Games Editor Menu. Since you want to edit, press (RETURN).

Good job! The preparation is over. You're now ready to create your new quiz.

Typing the Quiz

If you look at the screen, you'll notice that the program has laid out the five strings in their maximum size and placed the cursor at the beginning of the first string. You have just short of four lines for each string, each of which ends with a closing quotation mark, followed by a white rectangle.

The quotation mark is there to indicate the absolute maximum length the string may be; it is rare that your match will fill out the string. You may enter less than the maximum length by ending the string with a quotation mark. Then press (RETURN). In no case may you exceed the limit of the quotation mark. If you do, you will see this message:

```
***TOO LONG ERR
```

You will then have to retype the line. The last string on the screen is followed by a RETURN statement. This is the longest the last line, which contains the RETURN statement, may be without causing an error message.

Keep in mind that displaying these maximum string sizes and the RETURN statement on the last line are a way of reminding you of limitations and requirements. You will often find that you have more than enough room for your matches. And you may put the RETURN statement anywhere as long as it follows the last string used.

The title you want to appear on The Match Machine Menu is `ANIMAL CLASSIFICATION`. After you type the quiz, your screen will look similar to Figure 4-4.

Figure 4-4. *Animal Classification Quiz*

```
TYPE 'RUN 2' TO RETURN TO EDITOR
>9085 D1$="ANIMAL CLASSIFICATION,
ANIMAL&ORDER,WHALE&CETACEA,MONKEY
&PRIMATES,MOSQUITO&DIPTERA"
                                  "
 9086 D2$="DOG&CARNIVORA,KANGAROO
&MARSUPIALIA,ELEPHANT&PROBOSCEDEA
"
                                  "
 9087 D3$="EDENTATA,RATITAE"


                                  "
 9088 RETURN


                                  "
 9089 REM


                                  "
LAST STATEMENT MUST BE "RETURN"
RUN 2 <==COPY TO RETURN TO MENU
```

Here's how to add the Animal Classification quiz:

1. Press → to move the cursor just beyond the first quotation mark on the first line.

2. Now, type your first program line: first the title, then the column headings, then the matches themselves. Then press (RETURN).

For example, your first line could look like this:

```
9085 D1$="ANIMAL CLASSIFICATION,
ANIMAL&ORDER,WHALE&CETACEA,MONKEY
&PRIMATES,MOSQUITO&DIPTERA"
```

Since you didn't fill the allotted space in the string, be sure to include the final quotation mark after the last character in the string and then press (RETURN). You then see the BASIC prompt.

3. Press (←) (or (↓), if you have one on your computer) to move the cursor to the beginning of the next numbered line. Then press (→) to move the cursor just beyond the opening quotation mark and type the next line. The line should look like this:

```
9086 D2$="DOG&CARNIVORA,KANGAROO
&MARSUPIALIA,ELEPHANT&PROBOSCEDEA
"
```

4. Enter the rest of the quiz. Line 9087 contains two extraneous answers. Line 9088 is the data table terminator line. Line 9089, the REM statement, is merely a space filler since The Shell Games program requires that you use all five program lines (Shell Games Principle 8). Notice that the RETURN and REM statements are not preceded by a string name like D1$ and that they are not placed in quotation marks. To enter these lines, just type over the string name and quotation marks.

The final lines of the quiz look like this:

```
9087 D3$="EDENTATA,RATITAE"
9088 RETURN
9089 REM
```

You could have crowded almost the entire quiz into the first string, but you may want to come back at a later time and lengthen the title, add new matches (which must occur before

the extraneous answers), or make other changes that require increasing the amount of material. It is far easier to do this if you have left some room originally.

The next thing you would normally do would be to follow the suggestion at the bottom of the screen and type RUN 2 to return to The Shell Games Editor Menu where you can correct any mistakes you've made. But instead, type

```
RUN
```

to run the program and try out your quiz. If there are any problems, you should refrain from correcting them until you have read the next section on editing. After you have completed the quiz, select option 0, END THE GAME, from The Match Machine Menu. You'll see the BASIC prompt (>).

As you can see, The Shell Games Editor makes new quiz entry easier by finding the proper location for the next quiz, doing some of the typing for you, and providing end-of-string and don't-forget-the-RETURN-statement reminders. But it takes time for The Shell Games Editor to find the location of the next new quiz. If you are entering an entirely new set of quizzes all at once, delete the entire data table by typing

```
DEL 9010, 9999
```

You may then find it a lot faster not to use The Shell Games Editor. If you are creating all new quizzes, and you know on which lines the next quizzes will start, it often can be faster to enter the quizzes without the editor. Try entering new quizzes both ways and see which you prefer.

Editing an Existing Quiz

If you have been following along, you have a match-pair consisting of ELEPHANT and PROBOSCEDEA. It turns out, however, that *proboscidea* is incorrectly spelled in the example. (Kudos to the three people who noticed the misspelling!) There are several ways to correct this mispelling. One is to correct line 9086 using escape mode. You also can type

```
POKE 33,33
```

before listing the line to keep BASIC from adding a lot of extra spaces. Then type

```
TEXT
```

to go back to normal after copying the line. All much too much work.

Here's how you can use The Shell Games Editor to correct mistakes:

1. To enter The Shell Games Editor, type

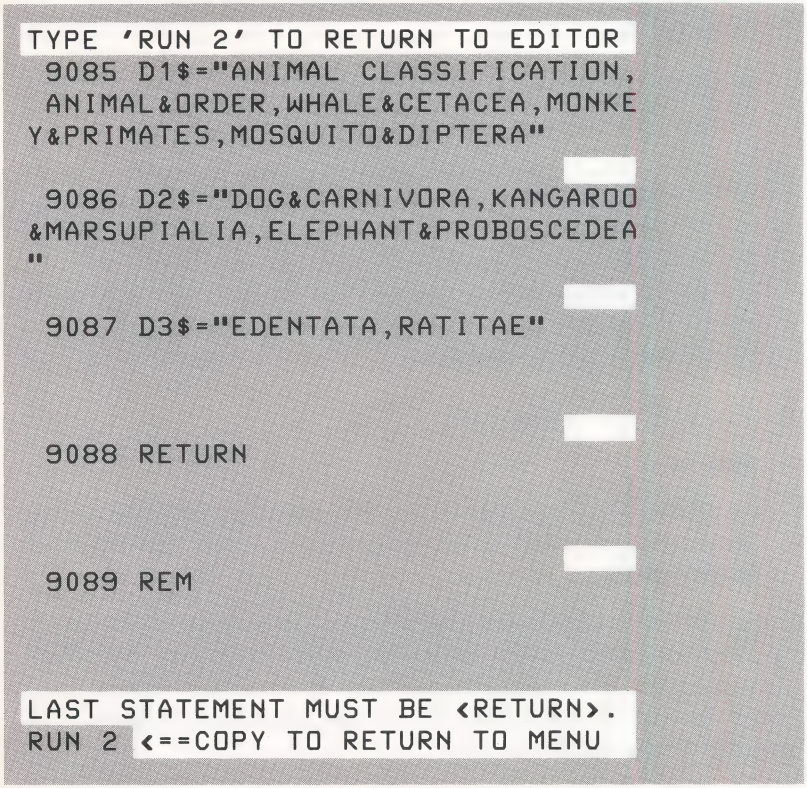
   ```
   RUN 2
   ```

   when the Integer BASIC prompt is displayed.

2. Select option 2, `EDIT CURRENT QUIZ`, from The Shell Games Editor Menu and press (RETURN).

3. You see the quiz you just added, and you are given the option of entering the editor or returning to the menu. Since you want to edit, press (RETURN). The screen now looks similar to Figure 4-5.

Figure 4-5. *Editing the Current Quiz*



The lines have been moved over so that no extra spaces are on the left. The cursor is at the beginning of the first string. You may enter as much text as will fit between the beginning of each string and the white rectangle. In fact, the cursor can be on top of the first block of the rectangle when you press (RETURN).

4. Next you'll correct the typing mistake.

   1. When you begin, the cursor will be on the first character of line 9085. Use (→) or (←) to move the cursor to the first character of line 9086. Now, press (→) until it's on the first E in PROBOSCEDEA. This copies over the line to the point where you've made the typing mistake.

   2. Type I to change the E to an I.

3. Press → until it is on top of the first space after the quotation mark. This copies over the remainder of the line.

4. Press RETURN.

   The new line has now replaced the original.

Remember: Whenever you correct a program line, you need to use → to move across the part of the line you don't want to change. When you do this you are making a new copy of the line to replace the original. If you don't use → to copy over the line, it is as if you had never typed the information in the first place.

5. You may now either type

   ```
   RUN 2
   ```

   or press ← till you get to the last line of text on the screen and copy over `RUN 2` by pressing →. You'll see The Shell Games Editor.

During the time that you are editing or entering your new quiz, you are not being controlled by The Match Machine. The Shell Games Editor has set up the window and printed helpful information on the screen, but you are essentially talking directly to the Apple computer.

If at any time while you are working on the quiz, the screen becomes cluttered or you get an error message that is covering up part of what you are doing, type

```
RUN 2
```

again and then select 2, `EDIT CURRENT QUIZ`, from The Shell Games Editor Menu. This will clear the screen and redisplay the lines you have already successfully entered.

Displaying a Quiz

Option 3 on The Shell Games Editor Menu, DISPLAY CURRENT QUIZ, lets you check to make sure that your quizzes are correct and in the right format. After selecting this option from the menu, your quiz is displayed in the normal way, except that now you have the option of instantly returning to The Shell Games Editor by pressing (ESC).

Try selecting this option and then going through the Animal Classification quiz to confirm that everything is working correctly. When you've finished with the quiz, don't press the (SPACE) bar. Instead, press (ESC). The Shell Games Editor is displayed.

Saving a Quiz

Once you have the quiz the way you want it, and if you want to keep the version of the program with the new quiz, you'll need to save what you've done on a disk.

To exit from The Shell Games Editor and save the revised version of The Match Machine, select option 7, END, from the menu. Be sure your backup copy of *The Shell Games* disk is in the disk drive, then type

```
SAVE THE MATCH MACHINE
```

The name you give to the revised program should continue to be THE MATCH MACHINE, since there isn't enough room on the disk for two copies of The Match Machine.

Searching for a Quiz

One of the prime reasons for creating The Shell Games Editor was the difficulty people encountered when trying to find a particular quiz somewhere in the middle of the data table. Listing 40 or 50 quiz questions to find the part with a typographical error can be most distressing. Two features of the Shell Games Editor were designed specifically to alleviate this problem.

First, in all The Shell Games, The Shell Games Editor keeps track of where you are in the program. To demonstrate this capability, try this example:

1. If you are in The Shell Games Editor, leave the editor by selecting option 7, END.

2. To display the current menu, type

   RUN

3. Select category 10 from the list of categories.

   If you are using an unaltered copy of The Match Machine, category 10 is called Animal Groups, Pt. 1. If your disk shows something different for category 10, it doesn't really matter. Just choose any category. You'll still be able to follow the example.

4. Now suppose that, in the midst of doing the quiz, you suddenly discover a typographical error. Being a perfectionist, you can't bear to continue with the quiz until the error is fixed. To correct the error, you first need to exit from the quiz. To do this, press (CONTROL)-(C) and then (RETURN).

5. To get to The Shell Games Editor, type

   RUN 2

   If you look at the menu, you see that The Shell Games Editor does indeed know that you are working on category 10 and is prepared to instantly display the program lines in that quiz. (It's all done with mirrors!)

Of course, there will be occasions when you do not want to edit the particular quiz you have been trying out. So The Shell Games Editor has a search option that allows you to find the quiz you want to work on. Here's how that search option works:

1. From The Shell Games Editor Menu, select option 4, SEARCH FOR QUIZ TO BE EDITED.

2. You may begin your search from one of three places: the first quiz in the data table, the current quiz, or the last quiz. When looking for a particular quiz, make your best guess as to which of these three locations is closest to the quiz you are searching for and select that option.

   Since option 3, `THE END OF THE DATA TABLE`, is the most fun, please select that now.

   When you choose that option, the editor goes forward from the last current quiz to the data table terminator. The line on your screen should be the data table terminator: a line number that is an even multiple of five with a RETURN statement.

3. To move back to the penultimate quiz, press ←. (Excuse that outbreak of sesquipedalianism, but as the life science experts in the audience were allowed to be in on the PROBOSCIDEA trick, it seems only fair to give the English wizards a chance to strike back.)

   To move back to the last quiz, press →.

   By using ← to move back through the quizzes and → to move forward, you may quickly find any quiz in memory. Note, however, that only the first line of each quiz is displayed. Experimentation proved this to be the most efficient scheme in terms of speed and clarity.

4. Once you have found the quiz you want to edit or display, press RETURN.

This takes you back to The Shell Games Editor Menu. The Shell Games Editor considers the quiz you were viewing at the time you pressed RETURN to be the new current quiz and is prepared to let you edit or display it.

Verifying the Data Table

As simple as the structure of the quizzes may now seem, it is quite easy when entering a lot of quizzes to make mistakes, such as not terminating each quiz with a RETURN statement or forgetting the data table terminator. Some of the errors created by these minor slips are difficult to understand. Not having a RETURN statement at the end of a quiz, for example, will cause the next quiz to occur twice!

Because it is the program's job to be concerned with such matters, verifying has been automated, and the editor will gladly do it for you when you select option 5, `VERIFY DATA TABLE FORMAT`, from The Shell Games Editor.

If there are any problems, the editor will tell you exactly what they are and help you to correct them, usually displaying the corrected lines and asking you to copy them by pressing (→).

After you have made the correction, The Shell Games Editor verifies that the correction was done properly, asking that you do it again if it is wrong. (Do not be frightened by this; it is very difficult to do it wrong. The check is there mainly to give you confidence that you definitely did it right.)

The only type of error that the editor cannot suggest a correction for is a missing RETURN statement. If you have forgotten the RETURN statement at the end of a quiz, The Shell Games Editor will tell you so and bring the program to the quiz that lacks the RETURN statement, leaving you in the editor. You should add a RETURN statement after the last string used and thus resolve the problem. Afterward, return to The Shell Games Editor and verify that the data table is correct.

When the problems are cleared up, the editor will give the data table a clean bill of health and announce that you should press (RETURN) to return to the menu.

Replacing an Old Quiz

Frequently, you will want to remove a quiz you no longer find useful and replace it with a quiz you have just created.

Here's how to replace one quiz with another:

1. Select option 4, SEARCH FOR QUIZ TO BE EDITED, on The Shell Games Editor Menu.
2. Select option 2, EDIT CURRENT QUIZ. Then, when the lines of the quiz are displayed, press (CONTROL)-(C).
3. Delete the lines that make up the old quiz by typing

   DEL <first line>, <last line>

   For example, if you wanted to delete a quiz that started on line 9010 and ended on line 9014, you would type

   DEL 9010, 9014
4. To return to The Shell Games Editor, type

   RUN 2
5. Create a new quiz, using option 1, ENTER A NEW QUIZ. The new, empty strings will be displayed and you may proceed to enter your new quiz. If this is your first time through The Shell Games Editor procedure, remember that you must actually enter a new quiz! You may not leave a gap in the data table! If you do, you are breaking two of The Shell Games principles: all Shell Game quizzes occupy exactly five lines, and all line numbers must be used.

Inserting and Deleting Quizzes

You can both insert and delete quizzes from The Shell Game's data tables.

Too Much? A much more straightforward way to make room for new quizzes is to delete the lines containing all of The Shell Games by typing

```
DEL 9010, 9999
```

Then create your own quizzes following the nine principles of The Shell Games and using escape mode to correct any errors.

Apple II and II Plus Owners: If you own an Apple II or an Apple II Plus with an Integer BASIC card, you have the Integer BASIC language built into your computer, and thus are not using the version of BASIC supplied on *The Apple Education Classics* disk.

The full Integer BASIC language system consists of five ROMs. The Renumbering Program, necessary for inserting and deleting questions, is in the fifth ROM, called Programmer's Aid #1. If you do not have this ROM, you may do everything in The Shell Games except insert and delete quizzes. If you want to find out whether you have this ROM, try the renumbering procedure. If the cursor goes away forever, or you find yourself looking at gibberish with an asterisk (*) prompt, you do not have the ROM necessary to do renumbering.

When you insert or delete a quiz, you need to renumber, by ones, all of the lines of the quizzes that come after the quiz you want to insert or delete. But before you do the renumbering, you need to make sure that each line in the data table is being used. If only four lines are used for a quiz, the next quiz's first line would suddenly become the last line of the previous quiz, creating chaos and ruin. This is the reason for Shell Games Principle 8: Each Shell Games quiz must use all five line numbers.

To make sure that all of your quizzes follow this rule, select option 6, VERIFY TABLE FORMAT TO ALLOW RENUMBER, before you start renumbering your program. When you select this option, the program checks to make sure all line numbers are used as well as doing the general error checking that occurs when you choose option 5, VERIFY DATA TABLE FORMAT. If any unused lines are discovered, the program will display the lines necessary to correct the problem and ask you to copy them.

Warning
It is vital that you verify the table format before attempting to renumber. Otherwise, you could be left with a frustrating and time-consuming unscrambling job.

Once you have verified that your quizzes include all the necessary line numbers, you can begin renumbering. But first, you must exit from The Shell Games Editor by choosing option 7, END.

Now, suppose that you want to insert a quiz before the fourth quiz, which currently is on lines 9025 to 9029. To make room for the new quiz, you need to start renumbering at line 9030. Here's how:

1. Type

```
CLR
```

This is the clear command. It clears the BASIC variable table.

2. Type the line number where the program portion will start, after renumbering. You want the renumbered portion of the program to start at line 9030. So type,

```
START = 9030
```

3. Next, choose the amount by which you want following line numbers to increase. You must always renumber The Shell Games by an increment of one, so type

```
STEP = 1
```

4. Now, you'll specify the line numbers that will be renumbered. In this example, you'll renumber all quizzes starting at line 9025. All quizzes will end long before line 9990, so that's where you will end the renumbering. So, the next two lines of the renumber sequence are

```
FROM = 9025
TO = 9990
```

5. The final step is a CALL statement. The CALL statement causes execution of the machine-language program that does the work necessary to get the renumbering done. When renumbering a quiz, the call address is always -10521. So, type

```
CALL -10521
```

While renumbering, your screen will look like Figure 4-6.

Figure 4-6. *Renumbering*

```
9064->9069
9065->9070
9066->9071
9067->9072
9068->9073
9069->9074
9070->9075
9071->9076
9072->9077
9073->9078
9074->9079
9075->9080
9076->9081
9077->9082
9078->9083
9079->9084
9080->9085
9081->9086
9082->9087
9083->9088
9084->9089
9085->9090
```

By the Way: The line numbers you see will differ, depending on the lines you are renumbering.

After about a minute, the BASIC prompt will reappear, along with the flashing cursor. Type

```
RUN 2
```

to go to The Shell Games Editor. Then choose option 1, ENTER A NEW QUIZ, and type your new quiz.

Deleting quizzes works very much the same way. Suppose that you want to delete the fifth quiz, which runs from line 9030 to line 9034, and then renumber the remainder of the quizzes to fill in the gap.

1. First, you need to delete lines 9030 to 9034. So, when the BASIC prompt is displayed, type

   ```
   DEL 9030, 9034
   ```

2. Next, give the clear command. Type

   ```
   CLR
   ```

3. Now, type the line number where the quiz will start, after renumbering. Type

   ```
   START = 9030
   ```

4. Next, choose the amount by which you want following line numbers to increase. Since you must always renumber The Shell Games by an increment of one, type

   ```
   STEP = 1
   ```

5. Now, you'll specify the lines that will be renumbered. You've already deleted lines 9030 through 9034, so the first line to get renumbered is 9035. All quizzes end long before line 9990, so that's where you will end the renumbering.

   ```
   FROM = 9035
   TO = 9990
   ```

6. The final step is to give the CALL statement. Type

   ```
   CALL -10521
   ```

The screen will show both the new and old line numbers. Then, after about a minute, you'll see the Integer BASIC prompt again.

Leaving the Editor

To leave The Shell Games Editor, select option 7, END, from The Shell Games Editor Menu. The editor reminds you to save your current program before proceeding.

Warning

Not saving your program will result in the loss of all the changes and additional quizzes you have entered, so don't forget to save!

The procedure for saving programs is described in "Saving a Quiz."

An Important Fringe Benefit

The usual way to exit from The Match Machine is to select option 0, END THE PROGRAM, from the list of categories, and then type RUN INDEX to return to The Shell Games Menu. This is not what you want to do if you have changed one or more quizzes, but have not yet saved. If you plan to change any of the quizzes, you should always enter The Shell Games Editor first by typing

```
RUN 2
```

before running the menu program (INDEX). When you do this, choose option 7, END, from The Shell Games Editor Menu. The type

```
SAVE <program name>
```

After you have saved your revised program, type

```
RUN INDEX
```

to return to The Shell Games Menu.

Better Safe Than Sorry: There are other more subtle protective measures taken by the editor to protect your work from harm, so it is a good idea to always enter The Shell Games Editor momentarily before working on any program, even if you will be entering quizzes without using the editor.

Don't Fall Word Lists

Don't Fall (one of the games on the *Elementary, My Dear Apple* disk) is actually a Shell Game. This means that, like the other Shell Games, the program is a *shell* into which you can put your own categories of words. This chapter explains how you can create your own Don't Fall word lists.

Creating Word Lists

Here's how to create your own Don't Fall word list:

1. Start up your backup copy of the *Elementary, My Dear Apple* disk. When the Elementary, My Dear Apple Menu is displayed, select option 4, `DON'T FALL`.
2. After the program is loaded and the disk drive stops whirring, press (CONTROL)-(C).
3. Type

   `TEXT`

   and press (RETURN) (ignore the flashing on the screen).
4. Type

   `LIST 9010, 9014`

When you list these lines, you see part of the Don't Fall program that creates a word list.

If you have the original program, here's what you will see:

```
9010 D1$="BIRDS,ROBIN,BLUEBIRD,EAGLE,
     CROW,TURKEY,SEAGULL,BLUEJAY,DUCK
     ,CHICKEN,GOOSE,HAWK,SWAN,ROADRUN
     NER,WREN,CARDINAL,SPARROW"

9011 D2$="FLAMINGO,PARROT,CANARY,PENG
     UIN,STORK,PARTRIDGE,PEACOCK,OSTR
     ICH,PIGEON,DOVE,WOODPECKER,OWL,P
     ELICAN,HUMMINGBIRD": RETURN

9012 REM

9013 REM

9014 REM
```

It doesn't matter if you find different lines in your program. All word lists have the same format.

Notice that these program lines look very similar to the program lines in The Match Machine quizzes, except that the word list program lines include individual words instead of matches. Specifically, each word list in the Don't Fall has two elements:

- The category name as it will appear on the Don't Fall Menu. (This may look strange on an Apple II or Apple II Plus because it has lowercase letters.)
- As many words in that category as you can think up and fit in five strings.

The category name and each word in the list is separated from the next word by a comma.

When typing lists of words, don't put in extra spaces. That means that, instead of typing

```
ROBIN, BLUEBIRD, CANARY
```

you should type

```
ROBIN,BLUEBIRD,CANARY
```

If you have a two-word name, like Flying Squirrel, type the space between the two words just as you normally would:

```
BADGER,FLYING SQUIRREL,FOX
```

The only rule you must observe when dividing up your word list among the strings is that breaks between strings must occur between words, where you would normally place a comma.

Don't Fall doesn't care whether you include the comma at the end of the string.

Creating Special Word Lists

If you played a version of Don't Fall that hasn't been altered, you may have noticed that the Spelling category had an asterisk (*) after it. A category marked with an asterisk means that the game doesn't end until you guess all the words in the category, in order. The Apple doesn't get to play Don't Fall when you choose a special category, but it will give a hint for each word you have to guess.

A special category (that is, a category with an asterisk after it) is constructed just like a normal word list, except that the name of the category is preceded by a pound sign (#). After each word, you can type an at sign (@) and the hint you want for that word. Follow the hint with a comma, just as in a normal word list. Notice that the longer the hints you give, the fewer words you can include.

For example, the Don't Fall special category called Spelling is constructed like this:

```
9070 D1$="#SPELLING,APPLE@A GREEN FRU
     IT,SALAMANDER@A SLIMY LIZARD"

9071 D2$="ICE CREAM@A COLD DESSERT,CH
     AIR@YOU SIT ON IT,LETTERS@POSTME
     N CARRY THESE,DOG@MAN'S BEST FRI
     END"

9072 RETURN

9073 REM

9074 REM
```

The pound sign in front of the category Spelling tells Don't Fall that this is a special word list. A word is separated from its hint by an at sign; each word-hint pair is separated from the next by a comma.

When the game asks you to guess a word from a special category, after it asks you to PLEASE GUESS A LETTER, instead of saying AND PRESS THE <RETURN> KEY, it will display the hint that's associated with the word. You have to guess all of the words in the category before you can stop playing.

By the Way: Teachers and parents may want to use these special categories to store weekly spelling lists, which students can use to quiz themselves while playing Don't Fall.

Modifying Don't Fall

Like Mr. Multiple and Professor True, Don't Fall has an option flag that enables you to change the way the program works. The name of the flag in Don't Fall is ANSWERALLINSEQUENCEFLAG and is located on line 9005. If this flag is set to YES, every quiz in the program will suddenly become a special word list, requiring the user to go through each and every word before leaving the category. If the flag is set to NO, no quiz will require all words guessed, even if you have put an asterisk before it. The flag comes set to OPTIONAL; in this state, quizzes must be done in full only if you have prefaced them with an asterisk.

This option could be useful in spelling quizzes, where you might want students to wander around at their will while learning, but then change all quizzes to mandatory when you want them to have more structured practice.

Here's how to change ANSWERALLINSEQUENCEFLAG:

1. Start up your backup copy of the *Elementary, My Dear Apple* disk. When the Elementary, My Dear Apple Menu is displayed, select option 4, DON'T FALL.
2. After the program is loaded and the disk drive stops whirring, press CONTROL-C.
3. Type

   TEXT

   and press RETURN (ignore the flashing on the screen).
4. Type

   LIST 9005
5. Then use escape mode to change OPTIONAL to YES or NO.

Editing Word Lists

The Shell Games Editor works exactly the same way when used with word lists as it does when used with The Match Machine. To use the editor with Don't Fall word lists, follow these steps:

1. When the Elementary, My Dear Apple Menu is displayed, select option 4, DON'T FALL.
2. After the program is loaded and the disk drive stops whirring, press CONTROL-C.
3. When the BASIC prompt (>) is displayed, type

   RUN 2

The editor should look fairly familiar! The only difference is that option 3—the option that lets you view your newly created list—is GO TO GAME, instead of DISPLAY CURRENT QUIZ as it is in the editor for The Match Machine.

Professor True and Mr. Multiple

This chapter will complete your exploration of *The Apple Education Classics* by giving you the ins and outs of the remaining Shell Games: Professor True and Mr. Multiple.

Inside Professor True

This section explains how you can add questions to Professor True's store of knowledge. But before attempting such a feat, you should have worked through the preceding chapters in this book. This is necessary because most of the things you will need to know to use Professor True are explained in Chapters 3 and 4.

Some Armchair Philosophy

Professor True is a highly flexible true/false quiz shell. As with Mr. Multiple, its reason for being is to teach, not to test. There is no way to use this quiz structure for graded examinations. You may, however, require that all questions be answered correctly before the user can exit the program, thereby creating a quiz on which all students will get 100 percent. (The way to do this is discussed in Chapter 2.)

There are two differences between Professor True and Mr. Multiple:

- First, there is the rather obvious difference of Professor True being a true/false quiz and Mr. Multiple being a multiple choice quiz.
- The other difference is that Professor True was designed for young people. The number of words you may have within a single problem is far fewer, the number of choices the student can make is by definition restricted to two, and there is an extensive amount of stroking and positive reinforcement. None of this, of course, restricts the game's use by older people.

In both programs, the color, sound, and game format, with little pieces of information being digested easily, keep the player's interest and attention.

Some Background

The basic structure of all The Shell Games is the same: The format of the quizzes is the same, and the same editor is used for each game.

The primary difference between Professor True and the Match Machine is that, in the Match Machine, the basic unit of information is a pair. In Professor True and Mr. Multiple, the basic unit is a word. There are some special symbols to be learned, but you will find things pretty much the same as in The Match Machine.

Here are the special symbols for Professor True:

Symbol	Meaning
Asterisk (*)	Indicates that the following question is true.
At Sign (@)	Signals the beginning of the explanation.
At Sign (@)	Signals the beginning of a new paragraph in the explanation.

A Real Example

Suppose you want a question in Professor True to look like Figure 6-1.

Figure 6-1. *Sample Question for Professor True*

```
THE U.S. TEMPORARILY OCCUPIED GRENADA
IN LATE 1983.
```

And you want to create an explanation that looks like Figure 6-2.

Figure 6-2. *Sample Explanation for Professor True*

```
THE U.S. OCCUPIED GRENADA FOR A
SHORT PERIOD IN THE LATER PART
OF 1983 TO PROTECT AMERICAN
CITIZENS OFFICIALLY LIVING ON
THE ISLAND.
```

First, you need to get The Shell Games Editor for Professor True. Here's how:

1. Select PROFESSOR TRUE from The Shell Games Menu.
2. When you see the Professor True introductory screen, press (CONTROL)-(C).
3. Now, type

   RUN 2

   You should see The Shell Games Editor.
4. To create a new problem (or question), select option 1, ENTER A NEW PROBLEM.

Before you actually begin typing the question, keep these points in mind:

- Because the answer is true, start the question with an asterisk. The asterisk must be the very first character of the question.
- When typing your question, you can end the first string when it is convenient. You may end anywhere as long as it is not in the middle of a word. Anywhere. For example, as far as the editor is concerned

```
9210 D1&="NOW IS THE TIME"
9211 D2$="FOR ALL GOOD MEN TO"
9212 D3$="COME TO THE AID OF THEIR"
9213 D4$="PARTY."
```

  is the same as

```
9210 D1$="NOW IS THE TIME FOR ALL GO
OD MEN TO"
9211 D2$="COME TO THE AID OF THEIR
PARTY."
```

  The program knows to add a space where each pair of strings joins so that you do not have to worry about having a space after TO or before COME.

- After you have typed the complete question, type an at sign (@) and begin typing the first paragraph of the explanation, again stopping when you begin to near the second quotation mark.
- If the answer has a second paragraph, type another at sign and continue typing.
- Explanations are strictly voluntary. If you do not want explanations, don't include them.
- When you are through, be sure to end with a RETURN statement.

Your final question should look something like this:

```
9210 D1$="*THE U.S. TEMPORARILY OC
CUPIED GRENADA IN LATE 1983.@THE US
OCCUPIED GRENADA FOR A SHORT PER
IOD"

9211 D2$="IN THE LATER PART OF 1
983 TO PROTECT AMERICAN CITIZENS
OFFICIALLY LIVING ON THE ISLAND."

9212 RETURN

9213 REM

9214 REM
```

Your Line Numbers May Be Different: The line numbers in your program may be different. Also, the breaks between strings need not duplicate the ones in the example, as long as those breaks do not occur in the middle of a word. Please note that the paragraph symbols do not have any spaces around them. If you wanted to indent the paragraphs, you could put three to five spaces after each at sign.

After you have completed typing the question, return to The Shell Games Editor Menu and display your problem by selecting option 3, DISPLAY CURRENT PROBLEM. If things don't appear as they should, check for the following:

- Do you have an asterisk as the very first character? If not, the program will think the answer is false.
- Do you have spaces surrounding an at sign? If so, your program will put a space on the screen.
- Did you put a RETURN statement at the end? If not, the program may display the wrong question or no question at all.

If something serious happens, such as the program won't run, go to The Shell Games Editor (type RUN 2) and select 5, VERIFY DATA TABLE FORMAT, from the menu to verify that the basic anatomy of your quiz is all right. If a problem is found, modify the program to correct it.

And that's all there is to it! Now you can exit The Shell Games Editor by choosing option 7 and then save this new version of the program, or you can want to enter more current questions and save an up-to-date version.

When you want to create a completely new true/false series, make a new backup copy of *The Shell Games* disk. Then, type

```
DEL 9010, 9999
```

to clear out the current quiz. Then you can start entering new questions without using The Shell Games Editor beginning on line 9010, or you can select `ENTER A NEW PROBLEM` from The Shell Games Editor Menu. Keep your questions, including the explanation, to around 40 to 50 words and type away.

Then run the game and enter The Shell Games Editor if you discover any problems. The original American History quiz of 40 questions took about 20 minutes to enter and another 15 minutes to edit. How long it took to write is another matter entirely.

Inside Mr. Multiple

If you have been following along program by program, you are about to find that the most sophisticated program in The Shell Game series is the easiest to create.

This section will show you how to create the question in Figure 6-3.

Figure 6-3. *Sample Question for Mr. Multiple*

```
YOU ARE PLAYING DICE AND HAVE ROLLED
23 SEVENS IN A ROW. THE ODDS OF YOUR
ROLLING A SEVEN AGAIN THIS TIME ARE:

1) BETTER THAN USUAL

2) USUAL

3) WORSE THAN USUAL
```

You'll also learn how to include the explanation in Figure 6-4.

Figure 6-4. *Sample Explanation for Mr. Multiple*

```
GAMBLERS WILL INSIST THAT HITTING A
SEVEN AGAIN IS ALL BUT IMPOSSIBLE.

MATHEMATICIANS WILL EXPLAIN THAT
EACH ROLL IS A SEPARATE ENTITY AND
THE ODDS ARE NO DIFFERENT FOR THE
24TH ROLL THAN THEY WERE FOR THE
FIRST.

PRAGMATISTS, HOWEVER, WILL
POINT OUT THAT THE ODDS ARE YOU WILL
CONTINUE THROWING SEVENS BECAUSE THE
ODDS ARE THERE IS SOMETHING SERIOUSLY
WRONG WITH THE DICE.
```

Here are the special symbols for Mr. Multiple:

Symbol	Meaning
Semicolon (;)	Signals the beginning of a possible answer that will be numbered and indented.
Asterisk (*)	Signals the correct answer. The asterisk is placed immediately after the semicolon.
At Sign (@)	Signals the start of the explanation (same as in Professor True).
At Sign (@)	Signals the start of a new paragraph in the explanation (same as in Professor True).
Semicolon (;)	Signals that an explanation will end with another list, also numbered and indented.

As you can see, the symbols, except the semicolon, are the same as those for Professor True. Although, the asterisk has a slightly different meaning. In Professor True, it means the question is true; here, it points out the right answer among all the possible answers.

Here's how to create a Mr. Multiple problem:

1. First select `MR. MULTIPLE` from The Shell Games Menu.
2. Then, when you are asked to type your name, press CONTROL-C and RETURN.
3. Go to The Shell Games Editor by typing

   ```
   RUN 2
   ```

4. Select option 1, `ENTER A NEW PROBLEM`.

When you start entering the question, keep in mind that you have much more information to include than in any of the other programs so you no longer have string space to burn. Fill each string to its capacity and do not forget the RETURN statement at the end of the question.

When you are done, your question should look similar to this:

```
9135 D1$="YOU ARE PLAYING DICE
AND HAVE ROLLED 23 SEVENS IN A RO
W. THE ODDS OF YOUR ROLLING A SEV
EN AGAIN THIS TIME ARE:"

9136 D2$=";*BETTER THAN USUAL;US
UAL;WORSE THAN USUAL@GAMBLERS WIL
L INSIST THAT HITTING A SEVEN AGA
IN IS ALL BUT IMPOSSIBLE.@"

9137 D3$="MATHEMATICIANS WILL EX
PLAIN THAT EACH ROLL IS A SEPARAT
E ENTITY AND THE ODDS ARE NO DIFF
ERENT FOR THE 24TH ROLL THAN"

9138 D4$="THEY WERE FOR THE FIRS
T.@PRAGMATISTS, HOWEVER, WILL POI
NT OUT THAT THE ODDS ARE YOU WILL
CONTINUE"

9139 D5$="THROWING SEVENS BECAUS
E THE ODDS ARE THERE IS SOMETHING
SERIOUSLY WRONG WITH THE DICE."
:RETURN
```

Enter The Shell Games Editor and display the question, clearing up any errors using the editor. Remember that if the screen gets messy, you can get a fresh start by typing RUN 2 and then selecting option 2, EDIT CURRENT PROBLEM.

As with the other Shell Games, if you want to keep this question in the quiz, exit using option 7 and then save the new version on your backup disk, by typing

```
SAVE MR. MULTIPLE
```

General Comments

With questions approaching 100 words in length, it can get pretty crowded inside Mr. Multiple strings. So keep your questions and answers as short as possible. You will have an easier time entering them and an easier time keeping your audience's attention.

Considering how few special symbols are associated with Mr. Multiple, the number of formats that you can create for your questions is quite amazing. It is worth exploring some of these formats. To do this, enter The Shell Games Editor. Then, select `SEARCH FOR CURRENT QUIZ` and pick question 1 as the current problem. Finally, select `DISPLAY CURRENT PROBLEM` to go through the program and examine each question. When you don't understand how a question was created, press (ESC) to go back to The Shell Games Editor and examine the questions in the editor.

Finishing Up

In this chapter, you'll learn how to make your own Shell Games disks, including how to rename programs.

Creating New Shell Games Disks

You now know how to replace, add, and delete quizzes on *The Shell Games* disk. The problem is that the disk only has enough room for a few new quizzes. If you want to create several new quizzes, you should create a completely new Shell Games disk. Using The Match Machine as an example, here's how to make a new Shell Games disk:

1. Make an additional backup copy of *The Shell Games* disk using either the DOS 3.3 COPYA program, the ProDOS Filer on the *ProDOS User's* disk, or the *Systems Utilities* disk.
2. Start up your new backup disk and select `THE MATCH MACHINE` from The Shell Games Menu.
3. When you are instructed to press the SPACE bar, press CONTROL-C.
4. When the BASIC prompt (`>`) is displayed, type

   ```
   DEL 9010, 9999
   ```

   This deletes The Match Machine's data table and the program lines with The Match Machine quizzes.

You now can use The Shell Game Editor to create new quizzes or, if you like, you can create quizzes without the editor.

If you want to add new sets of Professor True and Mr. Multiple questions to a new Shell Games disk, you don't need an additional disk. Just follow this procedure:

1. Select PROFESSOR TRUE or MR. MULTIPLE from The Shell Games Menu.
2. When you are asked to type your name, press (CONTROL)-(C) and the (RETURN). Then, delete the program's data table by typing

   ```
   DEL 9010, 9999
   ```

Then start adding new questions.

Changing Program Names

You may want to create a disk that contains several sets of Match Machine, Don't Fall, Mr. Multiple, or Professor True quizzes, each with a different name. Suppose, for example, that you want to create a set of Mr. Multiple problems called KUMQUATS. First, you need to delete the existing Mr. Multiple quiz by typing

```
DEL 9010, 9999
```

The deletion is necessary because there isn't enough room on the disk to store two different Mr. Multiple programs.

Now you can create a new set of quizzes. When you are ready to save your new quiz, be sure to give the quiz a new name. For instance, you might type

```
SAVE KUMQUATS
```

After you have created a new version of your Shell Game, you need to let the INDEX program know the new name so that it appears on The Shell Games Menu. You also can change the title that appears on the top of The Shell Games Menu.

To put the INDEX program into the computer's memory, type

```
RUN INDEX
```

Like the other Shell Games, INDEX keeps information in lines 9000 to 9999. Lines 9000 through 9002 of INDEX hold the title you want to appear at the top of The Shell Games Menu. The title is stored in three strings: D1$, D2$, and D3$.

Line 9003 contains SPACEFLAG, which comes set to one. SPACEFLAG causes a blank line to be printed after each name on the menu. If you create a disk with a lot of games on it, you may want the menu to be single-spaced. If you do, make SPACEFLAG equal to zero.

The names of the programs on the menu begin on line 9005. Each new program begins on a line number that is a multiple of five, and the program knows when it is finished reading the program names when it finds a new line with only a RETURN on it—just as with each of The Shell Games.

Each program name is alotted three 37-character strings: D1$ and D2$ hold the name of the program as you want it displayed on the menu. D3$ holds the name of the program as you saved it on the disk. As usual, you must have a RETURN statement after the last string. You need not have REM statements unless you want to reorder INDEX using the renumber program.

Here is an example of how you'd create a menu called Geography for Geniuses that includes one Shell Game called Rivers: Where They Go and How They Flow that you saved on disk with the name Rivers.

```
9000 D1$="GEOGRAPHY FOR GENIUSES"
9001 REM
9002 REM
9003 SPACEFLAG=1
9004 REM SPACEFLAG WILL CAUSE A BLANK
     LINE TO BE PRINTED AFTER EACH
     TITLE
9005 D1$="RIVERS: WHERE THEY GO
9006 D2$="AND HOW THEY FLOW"
9007 D3$="RIVERS"
9008 RETURN
9009 REM
```

Note that the words don't wrap around, that is, each program line corresponds to one line on the screen.

Summary

This appendix is a summary of the information presented in this book.

General Information

Modifying The Shell Games. Lines 9000 to 9009 of Professor True and Mr. Multiple contain flags that let you modify these programs. These lines also contain whatever text you want to appear on the quiz title page.

Starting Over. To prepare a Shell Game for an entirely new set of quizzes, first type

```
DEL 9010, 9999
```

to get rid of the old quizzes. Then start typing the new quizzes.

Using The Shell Games Editor. First select the program you want to work on from The Shell Games Menu. Then stop the program by pressing CONTROL-C and then RETURN. When you see the Integer BASIC prompt, type

```
RUN 2
```

Saving the Current Version. To save the current version of the game you are working on, type

```
SAVE <program name>
```

Returning to The Shell Games Menu. To return to The Shell Games Menu, type

```
RUN INDEX
```

Keep in mind that, when you type `RUN INDEX`, any modifications you have made to The Shell Games will be lost if you have not saved the current version.

Editing The Shell Games

To correct a program line, enter escape mode by pressing (ESC).

When you are in escape mode, here's how to move the cursor:

Press (I) or (↑)	To move the cursor up.
Press (M) or (↓)	To move the cursor down.
Press (J) or (←)	To move the cursor left.
Press (K) or (→)	To move the cursor right.

To exit from escape mode, press the (SPACE) bar.

Notice that escape mode lets you move the cursor without affecting what's on the screen or in the computer's memory.

When you are not in escape mode, you can press (→) to copy over text and (←) to erase text.

Replacing Text

To replace text in a Shell Game, follow these steps:

1. Press (ESC) to enter escape mode.
2. Press (↑) or (I) to move the cursor to the first digit in the line number of the line that contains the text you want to replace.
3. Exit from escape mode by pressing the (SPACE) bar.
4. Press (→) until the cursor is on the first character you want to replace.
5. Type the new characters over the old.

6. Press → to copy over the remainder of the line.
7. Press RETURN.

Deleting Text

To delete text in a Shell Game, follow these steps:

1. Press ESC to enter escape mode.
2. Press ↑ or I to move the cursor to the first digit in the line number of the line that contains the text you want to delete.
3. Exit from escape mode by pressing the SPACE bar.
4. Press → until it is on the first character you want to delete.
5. Press ESC to enter escape mode.
6. Press → or K so that the cursor moves over the portion of the line you want to delete.
7. Exit from escape mode by pressing the SPACE bar.
8. Use → to copy over the remainder of the line.
9. Press RETURN.

Inserting Text

To insert text in a Shell Game, follow these steps:

1. Press ESC to enter escape mode.
2. Press ↑ or I to move the cursor to the first digit in the line number of the line that contains the text you want to add to.
3. Exit from escape mode by pressing the SPACE bar.
4. Press → until the cursor is on top of the first character just past where you want to insert new characters.
5. Press ESC to enter escape mode.
6. Press ← or J to make room for the insertion. For example, if you want to insert three characters, including spaces, press ← or J three times.
7. Exit from escape mode by pressing the SPACE bar.

8. Insert your new text.
9. Use → to copy over the remainder of the line.
10. Press RETURN.

The Nine Shell Games Principles

The following is a list of the characteristics, or principles, shared by all The Shell Games:

1. The first question or quiz of each game must begin on line 9010.
2. Each question or quiz must occupy exactly five consecutive lines.
3. Every question or quiz has at most three kinds of BASIC statements: string statements, RETURN statements, and REM statements.
4. Each question or quiz can have up to five strings, which are named D1$, D2$, D3$, D4$, and D5$.
5. Each question or quiz must start with D1$.
6. While the use of strings other than D1$ is optional, the strings that you use must be in ascending order. That is, the sequence could be D1$, D3$, D5$, but not D1$, D5$, D3$.
7. Each question or quiz must have a RETURN statement after the last string used.
8. Each question or quiz must use all five assigned line numbers. Fill in unused lines with REM statements.
9. The last question or quiz in each game is the data table terminator, a RETURN statement where the first line of the next question or quiz would normally be.

The Match Machine

The Match Machine lets you create matching quizzes. Here's a list of the elements and symbols of a Match Machine quiz and a sample quiz.

Elements of The Match Machine

Element	Maximum
1 Quiz Title	34 Characters
1 Pair Column Headings	15 Characters Left Column Heading
	18 Characters Right Column Heading
Up to 10 Pairs of Matches	15 Characters Each Left Column Entry
	18 Characters Each Right Column Entry
4 Extraneous Answers (Optional)	18 Characters Each Entry

Special Symbols

Ampersand (&)	Connects the Matches
Comma (,)	Marks the End of a Match Set (or Title)

Example

To get this menu listing:

```
PHRASES AND THEIR MEANINGS
```

And this quiz:

```
PHRASE                              MEANING

KNOCK IT OFF-----------------DESIST
LIKE A SHOT------------------QUICKLY
BLUE SKY THE IDEA------------THINK ABOUT IT
UP IN ARMS-------------------INDIGNANT
BY DESIGN--------------------PURPOSELY
SPLIT HAIRS------------------QUIBBLE
                                    CHIDE
                                    WARM
```

Type this:

```
9010 D1$="PHRASES AND THEIR MEANI
NGS,PHRASE&MEANING,KNOCK IT OFF&DE
SIST,LIKE A SHOT&QUICKLY"

9011 D2$="BLUE SKY IT&THINK ABOU
T IT,UP IN ARMS&INDIGNANT,BY DESI
GN&PURPOSELY,SPLIT HAIRS&QUIBBLE,
CHIDE,WARM"

9012 RETURN

9013 REM

9014 REM
```

Don't Fall Word Lists

Don't Fall word lists use the same symbols as The Match Machine quizzes except that you won't need an ampersand because you won't have any matching pairs. Here's an example of how a Don't Fall word list might look inside The Shell Games Editor:

```
9010 D1$="BIRDS,ROBIN,BLUEBIRD,EAGLE,
CROW,TURKEY,SEAGULL,BLUEJAY,DUCK
,CHICKEN,GOOSE,HAWK,SWAN,ROADRUN
NER,WREN,CARDINAL,SPARROW"

9011 D2$="FLAMINGO,PARROT,CANARY,PENG
UIN,STORK,PARTRIDGE,PEACOCK,OSTR
ICH,PIGEON,DOVE,WOODPECKER,OWL,P
ELICAN,HUMMINGBIRD": RETURN

9012 REM

9013 REM

9014 REM
```

A special word list is a word list in which the user must guess every word before the game ends. To create a special word list, precede the name of the list with a pound sign (#). After each word, you can add an at sign (@) followed by a hint. Here's an example of how a special Don't Fall word list might look:

```
9070 D1$="#SPELLING,APPLE@A GREEN FRU
IT,SALAMANDER@A SLIMY LIZARD"

9071 D2$="ICE CREAM@A COLD DESSERT,CH
AIR@YOU SIT ON IT,LETTERS@POSTME
N CARRY THESE,DOG@MAN'S BEST FRI
END"

9072 RETURN

9073 REM

9074 REM
```

Professor True

Professor True lets you create true/false quizzes. Here's a list of the elements and symbols of a Professor True quiz and a sample question.

Elements of Professor True

Element	Maximum
1 Question	About 25 Words
1 Explanation	About 25 Words

Special Symbols

Symbol	Meaning
Asterisk (*)	Indicates that the following question is true.
At Sign (@)	Signals the beginning of the explanation.
At Sign (@)	Signals the beginning of a new paragraph in the explanation.

Example

To get this question:

```
THE LONG COUNT WAS BORN IN 1873
AND LIVED TO THE RIPE OLD AGE OF
105, THEREBY EARNING HIS NAME.
```

And this explanation:

```
THE LONG COUNT OCCURRED IN THE
DEMPSEY-TUNNEY HEAVYWEIGHT
CHAMPIONSHIP FIGHT, SEPTEMBER
22, 1927, WHEN THE REFEREE
ALLOWED TUNNEY TO REMAIN ON THE
CANVAS FOR 13 SECONDS WITHOUT
CALLING HIM OUT.

TUNNEY WENT ON TO WIN THE
CHAMPIONSHIP.
```

Type this:

```
9210 D1$="THE LONG COUNT WAS BOR
N IN 1873 AND LIVED TO THE RIPE O
LD AGE OF 105, THEREBY EARNING HI
S NAME.@THE LONG COUNT"

9211 D2$="OCCURRED IN THE DEMPSEY
-TUNNEY HEAVYWEIGHT CHAMPIONSHIP
FIGHT, SEPTEMBER 22, 1927, WHEN T
HE REFEREE"

9212 D3$-"ALLOWED TUNNEY TO REMA
IN ON THE CANVAS FOR 13 SECONDS W
ITHOUT CALLING HIM OUT.@TUNNEY WE
NT ON TO WIN"

9213 D4$="THE CHAMPIONSHIP.": RET
URN

9214 REM
```

Mr. Multiple

Mr. Multiple lets you multiple choice quizzes. Here's a list of the elements and symbols of a Mr. Multiple quiz and a sample question.

Elements of Mr. Multiple

Element	Maximum
1 Question 1 Explanation	Combined Total Words (Question and Explanation) About 90

Special Symbols

Symbol	Meaning
Semicolon (;)	Signals the beginning of a possible answer that will be numbered and indented.
Asterisk (*)	Signals the correct answer. The asterisk is placed immediately after the semicolon.
At Sign (@)	Signals the start of the explanation (same as in Professor True).
At Sign (@)	Signals the start of a new paragraph in the explanation (same as in Professor True).
Semicolon (;)	Signals that an explanation will end with another list, also numbered and indented.

Example

To get this question:

```
WHAT IS THE WORLD'S MOST POPULAR
FOOD?

1) BEEF

2) EGGS

3) RICE

4) MILK AND CREAM

5) POTATOES
```

And this explanation:

```
MILK IS THE WORLD'S FAVORITE BY FAR.
OVER 145 QUARTS PER PERSON PER YEAR
ARE CONSUMED.

HERE ARE SOME WORLD FAVORITES:

1) 120.2 LBS. OF POTATOES

2) 88.9 LBS. BEEF

3) 87.7 LBS REFINED SUGAR

4) 263 EGGS WEIGHING A TOTAL OF
      35.3 LBS.

5) 7.7 LBS. RICE
```

Enter this:

```
9210 D1$="WHAT IS THE WORLD'S MO
ST POPULAR FOOD?;BEEF;EGGS;RICE;*
MILK AND CREAM;POTATOES@"

9211 D2$="MILK IS THE WORLD'S FA
VORITE BY FAR. OVER 145 QUARTS P
ER PERSON PER YEAR ARE CONSUMED.@
HERE ARE SOME"

9212 D3$=" OTHER WORLD FAVORITES:
;120.2 LBS OF POTATOES;88.9 LBS O
F BEEF;87.7 LBS OF"

9213 D4$="REFINED SUGAR;263 EGGS
WEIGHING A TOTAL OF 35.3 LBS;7.
7 LBS RICE"

9214 RETURN
```

Bibliography

Every attempt was made to ensure accuracy in the questions used in The Shell Games. Whenever possible, at least three sources were used to confirm each answer. Where leading sources disagreed, the question was dropped.

For example, there is raging controversy over the name of the fastest land animal—some references claimed the cheetah, some the pronghorn antelope. So the question was broadened to include all animals, where there is no controversy—the swift is the swiftest. There were two possible other contenders: The peregrine falcon, which is said to have been measured at 82 MPH, and a strange bug called the botfly, said to hit speeds of up to 800 MPH, a high enough velocity to penetrate man like a bullet. Fortunately, there is no credence to this peculiar belief; no known insect can exceed 36 MPH.

The following is a partial bibliography of the references used in creating The Shell Games questions.

Bronson, William. *The Earth Shook, The Sky Burned.* Garden City, N.Y.: Doubleday & Co., 1964.

Burnam, Tom. *The Dictionary of Misinformation.* N.Y., N.Y.: Ballantine, 1975.

Encyclopedia Britannica. N.Y., N.Y.: Encyclopedia Britannica & Co., 1970.

Guiness Book of World Records. N.Y., N.Y.: Bantam Books, Inc., 1983.

Fireman, Judy. *TV Book.* N.Y., N.Y.: Workman Publishing Company, 1977.

Haley, Alex. *Roots*. Garden City, N.Y.: Doubleday & Co., 1975.

Hennessey, D.L. *Citizenship*. P.O. Box 281, Berkeley, Calif.: 1978.

Information Please Almanac. N.Y., N.Y.: Information Please Publishing, 1969.

O'Neill, John J. *Prodigal Genius, The Life of Nikola Tesla*. N.Y., N.Y.: Tartan Books, 1944.

Oxford English Dictionary. Oxford, England: Oxford University Press, 1978.

Sagan, Carl. *The Dragons of Eden*. N.Y., N.Y.: Random House, 1977.

Scientific American. Various issues.

Settel, Irving. *History of Radio*. N.Y., N.Y.: Grossett & Dunlap, Inc., 1967.

Shulman & Youman. *How Sweet It Was*. Shorecrest, Inc., 1966.

Wallechinsky, David, et. al. *The Book of Lists*. N.Y., N.Y.: Bantam Books, 1978.

Wallechinsky, D., et. al. *The People's Almanac*. Garden City, N.Y.: Doubleday, 1975.

The World Almanac. N.Y., N.Y.: Newspaper Enterprise Association, 1977.

Worth, Fred L. *The Trivia Encyclopedia*. N.Y., N.Y.: Bell Publishing, 1979.

Note on Lamont Cranston

Finally, you are due an explanation of the strange circumstances under which Lamont Cranston appears twice in the quiz, Secret Identities, in The Match Machine. Lamont Cranston is a world explorer who has given Kent Allard permission to use his name as a secret identity. The Shadow, in turn, is the secret identity of Lamont Cranston, who is really Kent Allard.

Index

A

ALLOWEXPLANACION-SKIPFLAG 17
ampersand (&) in The Match Machine 30
Apple II 26, 60
  editing keys 36, 37
Apple II Plus 26, 36, 60
Applesoft BASIC 22
Applesoft BASIC prompt 22
arrow keys 36
asterisk (*)
  in Professor True 75
  in Mr. Multiple 82
at sign (@)
  in Don't Fall 69
  in Professor True 75
  in Mr. Mutliple 82

B

backing up disks 2
BASIC 22
  line numbers 28
  REM statements 27
  reserved words 17
  RETURN statements 27
  string statements 27
booting 3

C

CALL 62
CLR 62
column headings in The Match Machine 32
comma (,) in The Match Machine 32
CONTROL-C 24
copying disks 2
correcting mistakes 33, 35-41
Cranston, Lamont 106
creating a Shell Games disk 87
cursor 36

D

Darts 3-8
data table 26
  verifying 58
data table terminator 28

deleting quizzes 59, 87
deleting text 38-39
design philosophy 21
disks, backing up 2
  copying 2
DISPLAY CURRENT QUIZ 55
Don't Fall 10, 67

E

EDIT CURRENT QUIZ 52
editing a quiz 35, 52-54
editing text
  deleting text 38-39
  inserting text 40-41
  replacing text 37-38
editing keys 36, 37
Elementary, My Dear Apple disk 3-12
END 64, 65
escape mode 35-41

F

flags
  Don't Fall 70
  Mr. Multiple 18
  Professor True 16-18
fractions 3

G, H

hangman 10
headings, column (in The Match Machine) 32
HELLO program 24

I

INDEX program 24, 65, 89
inserting quizzes 59
inserting text 40-41
Integer BASIC 22
Integer BASIC card 60
Integer BASIC prompt 22

J, K, L

← 33
Lemonade 8
line numbers 28

M

Match Machine 24-26, 29-35
  column headings 32
  format 32
  matches 32
  saving 41, 55
Match Machine quiz
  deleting 59
  displaying 55
  editing 35, 52
  entering 33-35, 45-51
  inserting 59
  replacing 59
  saving 55
  searching for 55
math quiz 9
mistakes, correcting 33, 35-41
Mr. Multiple 80-84
  flags 18
  format 82
MUSTANSWERALLQUESTIONSFLAG 17
MUSTRETESTWRONGANSWERFLAG 17

N, O, P

POKE 52
pound sign 69, 70
Professor True 73-79
  format 75
Professor True Flags 16-18
PROFSPEAKS 16
Programmer's Aid #1 60
prompt 22

Q

quiz title 30
quiz, searching for 55
quizzes
  entering 33-35, 45-51
  deleting 59, 87
  saving 41, 65

R

REM statement 27-28
remark 28
Renumbering Program 60
renumbering quizzes 62
replacing quizzes 59
replacing text 37
reserved words 17
RETURN statement 27, 28
RUN 2 52

S

saving quizzes 41, 65
searching for quiz to be edited 55
semicolon (;) in Mr. Multiple 82
Shell Games 11
  deleting 59
  displaying 55
  editing 35, 52
  entering 33
  inserting 59
  quitting 24
  replacing 59
  saving 18, 41, 55, 65
  searching for 55
Shell Games disk, creating 87-88
Shell Games Editor 43-65
Shell Games flags 15-18
Shell Games principles 26
SKIPSCØRINGFLAG 17
SKIPEXPLANACIONIFRIGHT-ANDSERFLAG 17
SPACEFLAG 89
spaces 31
special word lists 69
START 62
starting up 3
STEP 62
string statements 27
strings 27, 30
Superdarts 6-7
Supermath 9
&&&SYNTAX ERR 38

T

table format, verifying data 58
table, data 26
terminator, data table 28
text 52
  deleting 38-39
  inserting 40-41
  replacing 37-38
&&&TOO LONG ERR 48

U

uppercase letters · 25

V

VERIFY TABLE FORMAT TO ALLOW RENUMBER 60
verifying data table format 58

W

word lists, Don't Fall 67
words, reserved 17

X, Y, Z